

Universidade do Minho

Escola de Engenharia

Departamento de Informática

Análise e Interpretação de Imagem Médica
com o apoio de Agentes de *Software*

Filipe Miguel Maria Marreiros

Dissertação submetida à Universidade do Minho
para obtenção do grau de Mestre em Computação Gráfica e
Ambientes Virtuais, elaborada sob a orientação do
Professor Doutor Victor Manuel Rodrigues Alves

2006

Dedico este trabalho
à Lise com muito amor,
aos meus país
e aos meus irmãos.

“O conhecimento próprio leva-nos como que pela mão à humildade.”

(Josemaría Escrivá)

Agradecimentos

Agradeço ao meu orientador, Professor Doutor Victor Manuel Rodrigues Alves, que desde o início esteve presente para me auxiliar e demonstrou um espírito de companheirismo extraordinário.

À minha namorada Mourylise Heymer por todo o seu apoio e carinho. Foi ela a responsável pelo *design* da interface gráfica, incluindo criação de *icons*. Sem o seu apoio não teria encontrado, certamente, tanta motivação e tempo para realizar este trabalho.

Ao Luís Nelas, ao CIT-Centro de Imagiologia da Trindade e ao HGSA – Hospital Geral de Santo António, pelo fornecimento das imagens médicas utilizadas para testes e descrições sobre os processos de diagnóstico por imagem médica.

À Ana Nelas, pela revisão geral do texto.

E finalmente, agradeço a toda a minha família, que sempre me apoiou e incentivou ao longo de todos estes anos de estudo.

Resumo

A evolução dos equipamentos utilizados na radiologia, desde as unidades de Ressonância Magnética (RM) de 3-tesla e sistemas de Tomografia Computorizada (TC) de 64 cortes em simultâneo até à última geração de tecnologias híbridas como a Tomografia por Emissão de Positrões (TEP)/TC está rapidamente a produzir volumes de imagens que ameaçam superar a capacidade de interpretação dos radiologistas. Os procedimentos actuais de leitura estão a tornar-se inadequados para analisar entre 300 e 500 imagens de uma TC de rotina ao tórax, abdómen, ou pélvis, e são ainda menos adequados para as 1500 a 2000 imagens de uma angiografia de TC ou de um estudo de RM funcional. Contudo os programas de computador para visualização continuam a apresentar o mesmo esquema para analisar as imagens: imitando, basicamente, o processo manual de visualização de películas (como pode ser observado na maioria dos programas de visualização comerciais). Por outro lado, os sistemas baseados em agentes têm permitido um progresso significativo no desenvolvimento e análise de *software*. Os sistemas multiagente têm efectivamente revelado sucesso numa vasta variedade de áreas de investigação e aplicação.

O presente trabalho visa o desenvolvimento de um sistema multiagente que possibilite a criação de um sistema de diagnóstico *multidisplay* para imagem médica. A arquitectura multiagente desenvolvida é escalável quer em termos de área de visualização, ou em termos de agentes de conhecimento para diagnóstico assistido por computador e protocolos de afixação de imagens. Neste trabalho é apresentada uma abordagem para ultrapassar o problema de leitura das imagens médicas através da implementação de estratégias alternativas de navegação que levam a uma apresentação mais "inteligente" das imagens a qual determina um aumento exponencial da área de diagnóstico. A arquitectura multiagente desenvolvida e os seus agentes serão descritos em detalhe sendo ainda objecto de atenção, questões referentes à implementação do protótipo criado para validação e avaliação do sistema.

Área de Aplicação:

Imagem médica.

Palavras-chave:

Sistemas de visualização de imagem médica, sistemas baseados em agentes, sistemas *multidisplay*, diagnóstico assistido por computador, DICOM.

Abstract

The artefacts evolution used in the medical imaging practice, going from 3-tesla Magnetic Resonance (MR) units and 64-slice Computer Tomography (CT) systems to the latest generation of hybrid Positron Emission Tomography (PET)/CT technologies is producing a huge volume of images that threatens to overload the capacity of the interpreting radiologists. The current workflow reading practices are becoming inadequate for reviewing the 300 to 500 images of a routine CT of the chest, abdomen, or pelvis, and are less so for the 1500 to 2000 images of a CT angiography or functional MR study. However the image visualization computer programs continue to present the same schema to analyse the images. Basically, imitating the manual process of film visualization as can be found in most commercial medical viewers. On the other hand agent based computing has been hailed as significant break-through in software development and analysis. Multiagents systems are revelling success in a wide variety of research and application fields.

This work concerns the development of a multiagent system that enables a multidisplay medical image diagnostic system. The multiagent system architecture is a scalable one, either in terms of display area or in terms of knowledge agents for computer aided diagnostic and intelligent hanging protocol tasks. In this work it is presented an approach to overcome the medical imaging reading problem, by implementing alternative navigation strategies that leads to an "intelligent" hanging of images, which, in turn, determines an exponential increase of the diagnostic area. The developed multiagent architecture and its agents are described in detail, and implementation issues of a prototype created for assessment purposes are also object of attention.

Application Field:

Medical Imaging.

Keywords:

Medical image visualisation systems, agent based systems, multidisplay systems, computer aided diagnostic, DICOM.

Conteúdo

Agradecimentos	iv
Resumo.....	v
Abstract	vi
Conteúdo	vii
Lista de Figuras.....	x
Lista de Tabelas	xii
Notação e Terminologia.....	xiii
Capítulo 1	15
Introdução	15
1.1 Enquadramento	15
1.1.1 Visualizadores e Sistemas de Visualização de imagem médica	15
1.1.2 Sistemas multidisplay	16
1.1.3 Apresentação de imagens médicas	16
1.2 Motivação e Objectivos.....	17
1.3 Estrutura da Dissertação	17
Capítulo 2	19
Análise de dados médicos (Imagiologia)	19
2.1 Enquadramento dos sistemas utilizados na imagiologia	19
2.2 Estrutura hierárquica da norma DICOM	20
2.3 Tipos de modalidades estudadas.....	21
2.3.1 Ressonância Magnética	21
2.3.2 Tomografia Computorizada	22
2.3.3 Imagens RM e TC.....	23
2.4 Especificação do processo de análise de dados médicos (Imagiologia)..	24
Capítulo 3	27
Visualizadores e Sistemas de Visualização de Imagens Médicas.....	27
3.1 Soluções Existentes.....	27
3.1.1 Merge eFilm TM	27
3.1.2 DicomWorks	28
3.1.3 Rubo Medical.....	28
3.1.4 Osiris	28
3.1.5 ImageJ	29
3.1.6 MRIcro.....	29
3.1.7 Julius	29
3.2 Análise das Soluções Existentes.....	29
3.2.1 Ferramentas de Medição.....	30
3.2.2 Ferramentas de zoom.....	30
3.2.3 Ferramentas de controlo do centro e largura da janela.....	30
3.2.4 Estrutura hierárquica de informação do paciente	30
3.2.5 Processamento de imagem.....	31
3.2.6 Possibilidade de visualizar várias imagens da mesma série	31
3.2.7 Volume e Surface Rendering	32
3.2.8 Multidisplay.....	34

Capítulo 4	36
Sistema Multiagente	36
4.1 Sistemas Multiagente.....	36
4.1.1 Agentes.....	36
4.1.1.1 Agentes Reactivos.....	37
4.1.1.2 Agentes Cognitivos.....	37
4.1.2 Interações nas sociedades de agentes	38
4.1.2.1 Arquitectura do Quadro Negro.....	39
4.1.2.2 Troca de Mensagens.....	40
4.1.3 Especificação de sistemas multiagente.....	40
4.2 Especificação do Sistema	42
4.2.1 Objectivos Globais.....	42
4.2.2 Arquitectura.....	43
4.2.3 Estrutura Lógica.....	45
4.2.3.1 O Agente de Preparação de Dados, <i>apd</i>	46
4.2.3.2 O Agente Estação de Controlo, <i>aec</i>	46
4.2.3.3 Os Agentes Terminais de Visualização, <i>atv_i</i>	48
4.2.3.4 O Ambiente <i>qenv</i>	49
4.2.4 Comunicação.....	50
4.2.4.1 Mensagens do Agente Estação de Controlo para Quadro Negro	50
4.2.4.2 Mensagens do Quadro Negro para o Agente Estação de Controlo	50
4.2.4.3 Mensagens dos Agentes Terminais de Visualização para o Quadro Negro	51
4.2.4.4 Mensagens do Agente Estação de Controlo para os Agentes Terminais de Visualização	51
Capítulo 5	53
Protótipo do Sistema de Visualização de Imagens Médicas Multidisplay	53
5.1 Sistemas Multidisplay.....	53
5.2 Estrutura do protótipo	54
5.2.1 Estação de Controlo.....	55
5.2.2 Terminais de Visualização.....	56
5.3 Protocolo de Afixação	57
5.3.1 Cálculo das áreas dos Terminais de Visualização.....	59
5.3.2 Cálculo das áreas das séries.....	61
5.3.3 Cálculo das áreas das imagens	62
5.4 Navegação.....	64
5.4.1 Navegação pela Worklist.....	65
5.4.2 Navegação pelas Séries.....	65
5.4.3 Navegação pelos níveis do Ambiente Virtual	66
5.5 Ferramentas de análise	69
5.5.1 Translação (Pan)	69
5.5.2 Zoom	70
5.5.3 Controlo do centro e largura da janela.....	70
5.5.4 Inversão de cor.....	71
5.5.5 Imagem de referência	72

5.5.6	Informação sobre a Imagem.....	87
5.5.7	Rotação.....	88
5.5.8	Flip Horizontal.....	89
5.5.9	Flip Vertical.....	89
Capítulo 6		90
Implementação do Protótipo.....		90
6.1	Tecnologias de suporte.....	90
6.1.1	Linguagens de programação.....	90
6.1.2	Interpretação de dados DICOM.....	90
6.1.3	APIs Gráficas.....	91
6.1.4	API de comunicação.....	91
6.2	Mapeamento de texturas utilizando OpenGL.....	92
6.2.1	Criação e visualização das texturas.....	92
6.2.2	Substituição das texturas.....	94
6.2.3	Eliminação das texturas.....	94
6.3	Optimizações.....	95
Capítulo 7		97
Conclusões finais e trabalho futuro.....		97
7.1	Conclusões finais.....	97
7.2	Trabalho futuro.....	98
Bibliografia.....		100
Referências na Web.....		104

Lista de Figuras

Figura 1: Estrutura Hierárquica DICOM	21
Figura 2: Equipamento RM	22
Figura 3: Equipamento TC	23
Figura 4: Matriz de <i>voxels</i> de uma imagem.....	23
Figura 5: Imagem axial referenciada na imagem anatômica através de uma linha azul (A seta laranja indica a linha de referência).....	25
Figura 6: <i>Scrollbar</i> horizontal para navegação pelas imagens de uma série	31
Figura 7: Especificação do <i>Layout</i> utilizando o eFilm™	32
Figura 8: <i>Lookup Table</i>	33
Figura 9: <i>Surface Rendering</i>	33
Figura 10: Estrutura do Quadro Negro	39
Figura 11: Arquitetura do sistema multiagente	43
Figura 12: Índices de posicionamento dos terminais de visualização do sistema	47
Figura 13: Componentes do Sistema <i>Multidisplay</i>	54
Figura 14: Interface da estação de controlo	55
Figura 15: Interface de um terminal de visualização	56
Figura 16: Fotografia do sistema em funcionamento utilizando quatro terminais de visualização	57
Figura 17: Componentes de controlo do protocolo de afixação.....	58
Figura 18: Áreas atribuídas aos <i>displays</i>	59
Figura 19: Áreas atribuídas às séries a) apenas série S1, b) duas séries S1 e S2 distribuídas horizontalmente, c) duas séries S1 e S2 distribuídas verticalmente	62
Figura 20: Exemplo de uma distribuição de imagens dentro de uma série	63
Figura 21: <i>Worklist</i> do sistema	65
Figura 22: <i>Sliders</i> utilizados na navegação pelas séries	66
Figura 23: Nível dos terminais de visualização.....	66
Figura 24: Nível do terminal de visualização seleccionado	67
Figura 25: Nível da imagem seleccionada.....	68
Figura 26: Os vários níveis do ambiente virtual.....	68
Figura 27: Barra de ferramentas	69
Figura 28: Exemplo - translação	70
Figura 29: Exemplo - <i>zoom</i>	70
Figura 30: Exemplo - controlo do centro e largura da janela.....	71
Figura 31: Exemplo – inversão da cor.....	71
Figura 32: Exemplo – referência.....	72
Figura 33: Exemplo da intersecção de uma linha de referência com as linhas limite do plano da imagem	77
Figura 34: Representação dos dois vectores auxiliares <i>V1</i> e <i>V11</i>	79
Figura 35: Exemplo da uma linha de referência inserida na imagem de referência	81
Figura 36: Distância de um ponto a uma linha	82
Figura 37: Linha de referência que possui um ponto exterior e um ponto interior	83
Figura 38: Divisão do espaço e respectivos valores binários atribuídos.....	85
Figura 39: Intersecção da linha de referência com as linhas delimitadoras	86

Figura 40: Área para escolha do novo ponto limite da recta de referênciação .	87
Figura 41: Exemplo - Informação sobre a imagem	88
Figura 42: Exemplos – Rotação	88
Figura 43: Exemplo – <i>Flip</i> Horizontal	89
Figura 44: Exemplo – <i>Flip</i> Vertical	89
Figura 45: Sistema de visualização <i>multidisplay</i> em funcionamento	92

Lista de Tabelas

Tabela 1: Regras-ponte do sistema multiagente	44
Tabela 2: Acontecimentos (ou eventos) do agente de preparação de dados ...	46
Tabela 3: Acontecimentos (ou eventos) do agente estação de controlo.....	48
Tabela 4: Acontecimentos (ou eventos) dos agentes terminais de visualização	49
Tabela 5: Mensagens enviadas pelo agente da estação de controlo para o quadro negro	50
Tabela 6: Mensagens enviadas do quadro negro para o agente estação de controlo	50
Tabela 7: Mensagens enviadas dos agentes terminais de visualização para o quadro negro	51
Tabela 8: Mensagens enviadas do agente estação de controlo para os agentes terminais de visualização	52

Notação e Terminologia

Notação geral

A notação ao longo do documento segue a convenção apresentada a seguir:

- *Texto em itálico* – para palavras em língua estrangeira (e.g., Inglês). Também utilizado para dar ênfase a um determinado termo ou expressão.
- **Texto em negrito** – utilizado para realçar um conceito ou palavra no meio de um parágrafo.
- Parágrafo com espaçamento simples – para extractos de código.

Acrónimos

ACR	<i>American College of Radiology</i>
ANSI	<i>American National Standards Institute</i>
API	<i>Application Program Interface</i>
AVI	<i>Audio Video Interleave</i>
DICOM	<i>Digital Imaging and Communications in Medicine</i>
GUI	<i>Graphical User Interface</i>
HIS	<i>Hospital Information System</i>
HL7	<i>Health Level Seven</i>
NEMA	<i>National Electrical Manufacturers Association</i>
MCR	<i>Medical Clinical Records</i>
PACS	<i>Picture Archiving and Communications Systems</i>
PCE	Processo Clínico Electrónico
RIS	<i>Radiology Information System</i>
RM	Ressonância Magnética
RV	Realidade Virtual
SM	Sistemas Multiagente
TC	Tomografia Computadorizada

Símbolos Gerais e Abreviaturas

<i>E</i>	Sistema
<i>Cs</i>	Contexto do sistema
<i>apd</i>	Agente de Preparação de Dados
<i>aec</i>	Agente Estação de Controlo
<i>atv_i</i>	Agentes Terminais de Visualização
<i>qenv</i>	Ambiente do sistema

Capítulo 1

Introdução

Desde sempre diversas áreas do conhecimento científico, têm tentado auxiliar a medicina e a saúde em geral, sendo que, actualmente, a área das Ciências da Computação é uma das que contribui mais activamente para o auxílio dos médicos nas suas tarefas diárias. No entanto, este envolvimento é relativamente recente e foi possível graças aos avanços da Física, que permitiram a criação de equipamentos médicos com geração de dados em formato digital. Os dados gerados encontram-se em formatos diversos (e.g. sinais vitais, diagramas, tabelas, imagens), contudo, neste trabalho, apenas é abordado o formato imagem, ou seja, a área da medicina designada por imagiologia.

Tal como foi referido anteriormente, os grandes avanços da Física nas últimas décadas abriram caminho ao surgimento de uma quantidade muito vasta de diferentes tipos de equipamento, designados por modalidades de imagem médica. Exemplos bem conhecidos destas modalidades são RM (Ressonância Magnética), TC (Tomografia Computorizada), Raio-X e Ecografia entre outras.

As imagens geradas são geralmente armazenadas em PACS (*Picture Archiving and Communications Systems*) sendo utilizada a norma DICOM (*Digital Imaging and Communications in Medicine*) como formato de imagem e arquivo. Estes procedimentos encontram-se normalizados pelo “Manual de boas práticas de radiologia” publicado em diário da república despacho Nº258/2003 (2º série).

1.1 Enquadramento

Actualmente os meios complementares de diagnóstico médico assumem um papel preponderante. Estes são fundamentais para efectuar diagnósticos precisos, porque disponibilizam informação adicional de extrema importância, que pode não ser obtida recorrendo aos meios tradicionais. A imagem médica é, sem dúvida, um dos meios mais relevantes neste contexto.

1.1.1 Visualizadores e Sistemas de Visualização de imagem médica

As imagens podem ser analisadas através de sistemas de visualização de imagem médica ou visualizadores.

Existem inúmeras soluções (comerciais e não comerciais) disponíveis, contudo, constata-se que a generalidade destes programas de visualização de imagem médica continuam a apresentar o mesmo esquema para análise das imagens, imitando, basicamente, o processo manual de visualização de películas. Por outro lado, a evolução dos equipamentos utilizados na medicina, desde as unidades de Ressonância Magnética (RM) e Tomografias Computorizadas (TC) de 64 cortes simultâneos até à última geração de tecnologias híbridas como a Tomografia por Emissão de Positrões (TEP)/TC, está a produzir rapidamente volumes de imagens que ameaçam superar a capacidade de interpretação dos radiologistas. Implicando que os procedimentos actuais de leitura (utilizando os visualizadores) estão a tornar-se inadequados para analisar entre 300 a 500 imagens de um TC rotineiro ao peito, abdómen, ou pélvis, e ainda menos adequados para as 1500 a 2000 imagens de uma angiografia TC ou de um estudo de RM funcional.

1.1.2 Sistemas multidisplay

O surgimento de sistemas *multidisplay* convencionais deve-se à evolução quer dos sistemas operativos quer do *hardware* gráfico. Permite que, actualmente, os sistemas de visualização médica recorram a esta característica, para aumentar consideravelmente a área disponível para análise. Contudo a maioria dos visualizadores encontra-se limitado pelo que o sistema operativo e o *hardware* têm a oferecer. Normalmente é possível utilizar apenas 2 monitores, mas existem soluções que permitem utilizar 8 [WEB-REALTIMESOFT'05] ou mais monitores ao mesmo tempo utilizando apenas um PC.

Alternativamente é possível produzir sistemas *multidisplays* utilizando *clusters* de PCs. Este é a situação mais comum, nas aplicações de realidade virtual, que utilizam sistemas *multidisplay*. São sistemas bastante mais escaláveis e flexíveis, pois não estão limitados a um conjunto restrito de *displays*¹ e permitem, em alguns casos, que exista uma distribuição da carga de processamento pelas várias máquinas do *cluster*.

1.1.3 Apresentação de imagens médicas

O principal desenvolvimento, realizado na imagiologia, tem sido no processamento de imagem, sendo negligenciadas as formas de apresentação das mesmas. Este facto pode ser facilmente constatado, observando os visualizadores existentes actualmente. Dos poucos trabalhos publicados sobre apresentação de imagens médicas, temos de destacar o trabalho desenvolvido

¹ Optou-se pela utilização do termo *display* em Inglês, por não ter sido encontrado, na língua portuguesa, uma tradução equivalente que permitisse reflectir de uma forma clara e sucinta o termo em causa. Assim, neste contexto, este refere-se a todos os dispositivos que possibilitam a visualização de informação gerada computacionalmente. Exemplos típicos de *displays* são monitores e superfícies onde são projectadas imagens.

pela Simon Fraser University [VAN DER HEYDEN'98a], [VAN DER HEYDEN et al.'98b], [VAN DER HEYDEN et al.'99a], [VAN DER HEYDEN et al.'99b], [KUEDERLE et al.'01a] e [KUEDERLE et al.'01b]. O objectivo principal destes trabalhos, consiste no estudo das melhores formas de apresentar imagens RM num único monitor. Em [VAN DER HEYDEN et al.'99b] é tentado emular um ecrã de luz (utilizado para visualização de películas), utilizando várias técnicas para superar o *screen real state problem*. Este problema pode ser descrito como o problema de apresentar informação dentro do espaço disponível de um monitor.

1.2 Motivação e Objectivos

A maioria dos sistemas de visualização de imagens médicas actuais, exploram exaustivamente técnicas de análise de imagem, contudo negligenciam a forma como estas são apresentadas. A área disponível de visualização das imagens destes sistemas é directamente dependente da área do ambiente de trabalho, disponibilizada pelo sistema operativo utilizado.

Estas observações deram a motivação necessária para o desenvolvimento um sistema de visualização de imagens médicas *multidisplay*. Este possibilita a visualização das imagens médicas nos vários *displays* do sistema, aumentando consideravelmente a área de trabalho dos utilizadores (e.g. Radiologistas), auxiliando-os na análise e navegação das imagens médicas. Desta forma superando as limitações da maioria dos sistemas de visualização actuais.

Foi dada especial atenção aos mecanismos e ferramentas de navegação, disposição (protocolo de afixação) e análise das imagens presentes no sistema, com o objectivo de facilitar a interacção e tornar estas ferramentas uma mais valia significativa para o processo global de análise.

Outro objectivo deste trabalho consiste no desenvolvimento de um sistema escalável, fácil de montar e barato, recorrendo à tecnologia de multiagentes, para suporte do sistema global.

1.3 Estrutura da Dissertação

A dissertação encontra-se organizada em sete capítulos: Introdução, Análise de dados médicos, Visualizadores e Sistemas de Visualização de Imagens Médicas, Sistema Multiagente, Protótipo do Sistema de Visualização de Imagens Médicas *Multidisplay*, Implementação do Protótipo e Conclusões finais e trabalho futuro.

No capítulo 1 é enquadrado o trabalho e exposta a motivação que deu origem a esta dissertação, bem como os objectivos que se pretendem alcançar com a mesma. É ainda apresentada a estrutura da dissertação.

Seguidamente, no capítulo 2, é apresentado o processo de análise de dados médicos na imagiologia. Numa primeira fase é apresentado o enquadramento dos sistemas utilizados na imagiologia, a estrutura hierárquica da norma DICOM e quais os tipos de modalidades médicas estudadas e no final é especificado o processo de análise de dados médicos.

No capítulo 3 são apresentados e analisados os visualizadores e sistemas de visualização de imagem médica existentes.

O sistema multiagente desenvolvido é descrito no capítulo 4. Inicialmente é fornecido um enquadramento sobre os sistemas multiagente e de seguida é apresentada a especificação do sistema.

O protótipo do sistema de visualização de imagens médicas *multidisplay* é apresentado no capítulo 5. Primeiramente é fornecido um enquadramento sobre os sistemas *multidisplay*, seguindo-se a apresentação da estrutura do protótipo e a descrição das funcionalidades do sistema, i.e. protocolo de afixação, navegação e ferramentas de análise.

No capítulo 6 são descritas as considerações tomadas para a implementação do protótipo. São apresentadas, nomeadamente, as tecnologias de suporte, o mapeamento de texturas utilizando OpenGL e as optimizações efectuadas.

Finalmente, no capítulo 7, são apresentadas as conclusões finais da dissertação e apontadas direcções a tomar para trabalho futuro.

Capítulo 2

Análise de dados médicos (Imagiologia)

O processo de análise de dados médicos depende de um conjunto de sistemas burocráticos, informáticos, etc. existentes dentro das instituições médicas, de todos os equipamentos de aquisição de imagem médica (modalidades) e obviamente dos profissionais de saúde. Para podermos especificar o processo de análise efectuado pelos clínicos é então fundamental conhecer primeiro todas estas componentes. Após esta introdução, pode ser especificado o processo de análise de dados médicos em imagiologia, com base em informação fornecida pelos especialistas. Esta especificação é fundamental para o desenvolvimento de sistemas de visualização que permitam facilitar o processo de análise.

Nas secções seguintes serão introduzidos: o enquadramento dos sistemas utilizados na imagiologia, a apresentação da estrutura hierárquica da norma DICOM, a descrição das modalidades estudadas e, finalmente, será apresentada a especificação do processo de análise clínico, com base em dados recolhidos junto dos especialistas.

2.1 Enquadramento dos sistemas utilizados na imagiologia

De acordo com [BRUCE'03] a imagem digital e o uso de computadores para processar imagens radiológicas foi introduzido nos anos 70. Para gerir as imagens digitais num arquivo electrónico, a base de dados utiliza um conjunto de campos, que se encontram num cabeçalho, podendo assim identificar as imagens. Exemplos são: nome do paciente, tipo de estudo, número do estudo, etc. Isto implica que, para aceder correctamente à informação através da base de dados, é necessário que o cabeçalho seja comum às várias imagens, ou seja, deve seguir as regras definidas numa norma.

Este facto introduziu grandes dificuldades no início da utilização destas tecnologias. O problema principal consistia no seguinte: cada modalidade (tipo de exame por exemplo: TC, RM) possuía um cabeçalho com campos distintos e organizados de formas diferentes. Existiam ainda casos em que modalidades iguais mas de fabricantes diferentes tinham cabeçalhos distintos.

Isto despoletou o desenvolvimento de uma norma que contemplasse a transmissão e armazenamento das imagens médicas digitais. Em 1983 a ACR (*American College of Radiology*) [WEB-ACR'05] e a NEMA (*National Electrical Manufacturers Association*) [WEB-NEMA'05] formaram um comité conjunto para desenvolver uma norma universal e assim surgiu o DICOM (*Digital Imaging and Communications in Medicine*) [WEB-DICOM'05].

O grupo de trabalho DICOM tinha por principal objectivo organizar os cabeçalhos das imagens médicas, independentemente dos fabricantes e das modalidades. Actualmente, a norma continua a evoluir, contemplando um número crescente de novas modalidades como a dermatologia, endoscopia, etc.

Após a criação desta norma foi possível, de uma forma eficiente, criar repositórios de imagens médicas designados por PACS (*Picture Archiving and Communications Systems*). O PACS é utilizado directamente com o DICOM para o envio de imagens para a base de dados, efectuar *queries* e aceder às imagens que se encontram armazenadas.

Existem ainda dois sistemas cruciais. São eles: o RIS (*Radiology Information System*) e o HIS (*Hospital Information System*). O RIS, tal como nome indica, é o sistema de informação do serviço de Imagiologia/Radiologia. Este trata toda a informação relativa aos exames, desde a marcação, execução e facturação. Quanto ao HIS, o sistema de informação do hospital, trata de toda a informação hospitalar, como por exemplo: salários, *stocks*, episódios, etc., possuindo ligação ao RIS.

O RIS e a maioria dos HIS possuem a sua própria norma de informação conhecida por HL7 (*Health Level Seven*) [WEB-HL7'05] uma norma ANSI (*American National Standards Institute*) [WEB-ANSI'05]. Esta norma permite identificar pacientes, ordens de processos, armazenar relatórios, etc., mas não pode gerir informação DICOM.

Finalmente os especialistas, para além das imagens médicas, necessitam também do PCE (Processo Clínico Electrónico) e dos MCR (*Medical Clinical Records*). Nestes processos e relatórios é possível consultar toda a informação clínica, desde relatórios médicos, imagens, resultados de análises, etc.

2.2 Estrutura hierárquica da norma DICOM

A norma DICOM define uma estrutura hierárquica em forma de árvore, que contempla vários níveis (figura 1). Como pode ser observado, o nível inferior é o Nível das Imagens, onde podem ser encontradas as imagens obtidas através dos equipamentos médicos. O nível seguinte é o Nível das Séries, onde as imagens são agrupadas. Na maioria das modalidades as imagens numa série são relacionadas; por exemplo, num TC estas correspondem normalmente a uma aquisição ao longo do corpo do paciente. O Nível dos Estudos agrupa estas séries. Finalmente, o Nível do Paciente contém todos os estudos de um paciente.

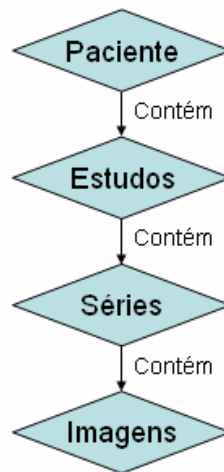


Figura 1: Estrutura Hierárquica DICOM

2.3 Tipos de modalidades estudadas

As modalidades que estamos a contemplar neste estudo são as RM e as TC. Este facto aplica-se a todo o trabalho, uma vez que existem secções onde a utilização de outras modalidades pode não ser aplicável. Não serão exaustivamente abordados os princípios físicos subjacentes à aquisição deste tipo de modalidades, contudo será apresentado um breve historial e a forma como são utilizadas.

2.3.1 Ressonância Magnética

Inicialmente, a RM (Ressonância Magnética) designava-se por RMN (Ressonância Magnética Nuclear). O termo Nuclear foi abandonado devido a conotações negativas e incorrectas com a energia nuclear. Em 1946, nos Estados Unidos, dois investigadores independentemente descreveram o fenómeno designado NMR (RMN) [BLOCH et al.'46] e [PURCELL et al.'46] e, posteriormente, receberam o prémio Nobel pelo seu trabalho. Estes investigadores foram Felix Bloch da Stanford University e Edward Purcell da Harvard University, e descobriram que certos núcleos, quando colocados num campo magnético, absorviam energia na gama das rádiofrequências do espectro electromagnético e reemitiam esta energia quando o núcleo voltava ao seu estado original. Este fenómeno foi designado por RMN: Ressonância porque a frequência depende directamente dos campos magnético e de rádiofrequência, Magnética porque um campo magnético é necessário e Nuclear porque só o núcleo de certos átomos reagem dessa forma.

Mas foi apenas em 1978 que Mansfield apresentou a primeira imagem de um tórax humano resultado da sua investigação [MANSFIELD&MAUDSLEY'76a] e [MANSFIELD&MAUDSLEY'76b]. Seguiram-se inúmeras séries de experiências e assim se iniciou a aplicação destas técnicas em

seres humanos. Também foi nos finais da década de 70 e inícios da década de 80 que empresas como Picker (actualmente Phillips) começaram o desenvolvimento destes sistemas para uso em aplicações clínicas.

Actualmente, o equipamento de aquisição RM é tecnologicamente muito avançado. O paciente deita-se numa mesa que se desloca para a zona de aquisição de imagem, controlada pelo operador do equipamento. A figura 2 apresenta um sistema actual.



Figura 2: Equipamento RM

No interior do cilindro encontra-se um magneto que irá produzir o campo magnético. As variações dos núcleos serão registadas por pequenos sensores que fornecerão os dados para obtenção da imagem digital.

2.3.2 Tomografia Computorizada

O primeiro sistema TC (Tomografia Computorizada) também designado por TAC (Tomografia Axial Computorizada ou Tomografia Assistida por Computador) foi inventado por Godfrey Newbold Hounsfield do EMI Central Research Laboratories, fazendo uso de raios-X. No entanto, independentemente, Allan McLeod Cormack da Tufts University inventou o mesmo processo. Por esse facto, os investigadores receberam conjuntamente o prémio Nobel da medicina em 1979. O TC pode ser visto como uma extensão ao convencional raio-X, isto porque efectivamente em vez de apenas uma emissão de raio-X, no TC são feitas n-emissões em direcções diferentes à volta do paciente.

Actualmente o equipamento de aquisição TC é tal como o RM, tecnologicamente muito avançado. Também como no caso do RM o paciente deita-se numa mesa que se desloca para a zona de aquisição de imagem, controlada pelo operador do equipamento. A figura 3 apresenta um sistema actual.

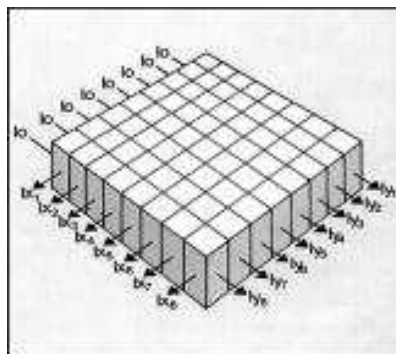


Figura 3: Equipamento TC

No interior do cilindro (*gantry*) encontra-se um conjunto elevado de detectores de raio-X e, no seu interior, um emissor que se move em círculos à volta do paciente. A combinação das detecções destas emissões de raio-X, após atravessar o corpo do paciente, é a responsável pela obtenção da imagem digital.

2.3.3 Imagens RM e TC

As imagens digitais, geradas pelos dois tipos de modalidades, possuem propriedades comuns, i.e., fundamentalmente são um conjunto de imagens, geralmente separadas uniformemente em intervalos espaciais, em que cada passagem é realizada num período de tempo reduzido para evitar movimentos. O conjunto destas imagens representa uma determinada área anatómica de estudo e pode ser utilizado para efectuar uma reconstrução tridimensional. Uma vez que o conjunto de imagens representa, efectivamente, um volume. Como possuímos um volume, cada valor existente na matriz da imagem pode ser interpretado como um *voxel*. Um *voxel* é o elemento base do volume, e é comparável ao *pixel* se possuímos uma área 2D e não um volume 3D. Os *voxels* podem ser interpretados geometricamente de várias formas, podem ser considerados como esferas centradas nos vértices de cada cubo elementar, mas geralmente são vistos como cubos que possuem um volume definido e associado a si um valor constante. A figura 4 apresenta graficamente uma matriz onde se encontram distribuídos os *voxels* de uma imagem digital.

Figura 4: Matriz de *voxels* de uma imagem

Juntando todas estas matrizes/imagens poderemos posteriormente efectuar reconstruções 3D. Contudo, os médicos radiologistas são especialistas em efectuar reconstruções mentais 3D navegando pelas imagens 2D. Devido a este facto e porque é bastante mais complexo parametrizar as imagens 3D, provocando um aumento do tempo despendido na análise quando da utilização destas técnicas 3D, implica que, numa esmagadora maiorias dos casos, se constate que, apesar de possuímos um conjunto de imagens representando um volume que pode ser reconstruído, apenas as imagens 2D são usadas para o diagnóstico médico.

2.4 Especificação do processo de análise de dados médicos (Imagiologia)

Tal como referido anteriormente, a especificação do processo de análise de dados médicos na imagiologia, baseou-se nas descrições dos clínicos envolvidos no processo, sendo os médicos radiologistas os principais contribuintes. Relembrando, as modalidades que estamos a contemplar são as RM e as TC, isto porque alguns dos procedimentos apresentados de seguida podem não se enquadrar utilizando outras modalidades.

Das descrições recolhidas foram identificados os seguintes procedimentos:

- 1. Escolha do exame:** No início o médico tem de escolher um exame que se encontra geralmente no PACS. Contudo a escolha do exame depende da informação existente no RIS, isto porque é aqui que a lista de trabalho (*worklist*) de exames a tratar se encontra disponível. Esta lista possui vários campos que facilitam a escolha do médico, por exemplo urgência, especialidade, tipo de exame, etc.;
- 2. Requisição de informação adicional:** Em muitos casos o médico necessita de mais informação para poder analisar o exame. Por exemplo ele poderá solicitar o PCE do paciente, exames com patologias semelhantes, etc. É de referir que a ordem pela qual este procedimento se encontra apresentado neste trabalho não é necessariamente o utilizado, por exemplo o médico poderá apenas requerer informação adicional no final da análise.
- 3. Ajuste de parâmetros das séries:** Após a escolha do exame o médico efectua um conjunto de manipulações que geralmente alteram todas as imagens de uma determinada série para as poder visualizar melhor, permitindo assim, seleccionar as imagens que procura. Normalmente estas operações consistem em alterar o centro e a largura da janela (*wc-ww*). Estes parâmetros afectam o brilho e o contraste das imagens. Quando o médico encontra uma solução razoável, para o seu caso particular, este poderá posicionar-se na zona de análise, e.g., um conjunto restrito e contínuo de imagens;

- 4. Referenciação através da imagem anatómica:** Em alguns exames especiais o médico necessita de uma referenciação entre as imagens axiais e a correspondente posição na imagem anatómica (descrita na norma DICOM como *Localizer*). Esta imagem anatómica na modalidade TC e.g. encontra-se na primeira série, que apenas contém a mesma. A imagem representa de forma global a área em estudo, como apresentado na figura 5.

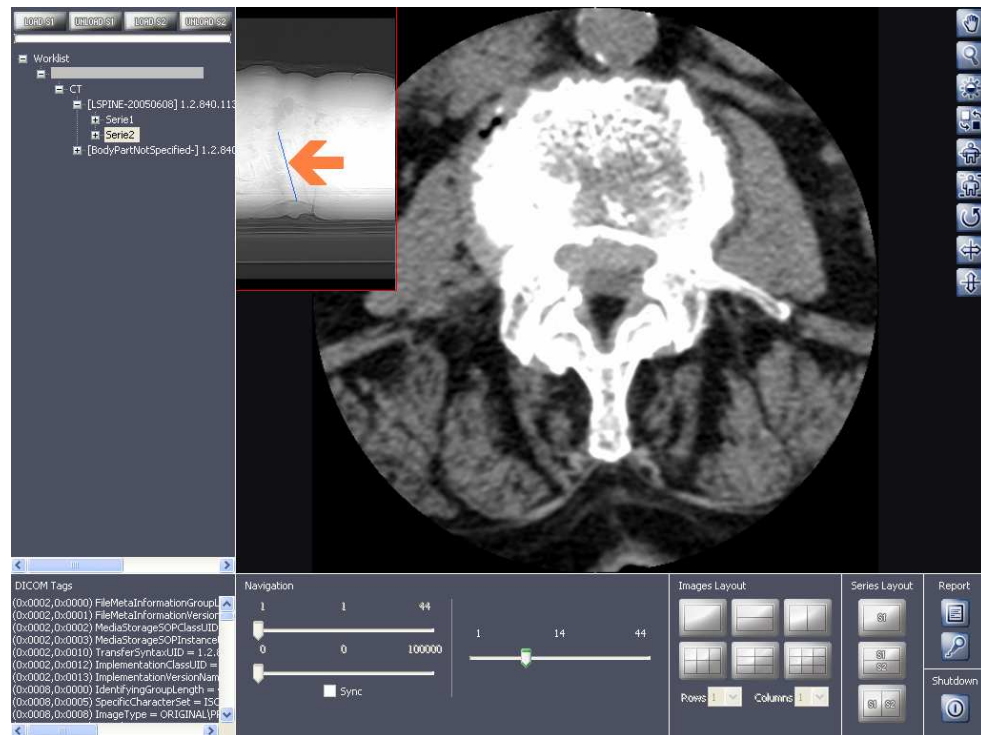


Figura 5: Imagem axial referenciada na imagem anatómica através de uma linha azul (A seta laranja indica a linha de referenciação)

Esta referenciação é necessária porque existem zonas anatómicas semelhantes e o médico, ao visualizar a imagem, pode não se conseguir localizar. Um destes casos acontece nos exames da coluna vertebral, devido ao facto das vértebras serem idênticas; o médico não consegue saber qual a vértebra que está a visualizar, mas utilizando a imagem anatómica e possuindo como referência uma linha que corresponde à posição da imagem seleccionada, este consegue facilmente localizar-se;

- 5. Análise individual das imagens:** Depois do médico identificar as imagens relevantes para o diagnóstico, efectua uma série de operações sobre as mesmas. Exemplos são medições de zonas anatómicas, efectuar *zoom* para melhor visualizar zonas de interesse, ajustar melhor o centro e a largura da janela wc-ww, etc.;

- 6. Executar relatório:** Depois da análise das imagens o médico executa o relatório que pode ser feito de várias formas, por exemplo: ditando utilizando reconhecimento de voz, escrevendo utilizando modelos de apresentação, etc.;
- 7. Anexar dados ao relatório:** Anexar imagens, *email*, *fax*, etc. ao relatório;
- 8. Terminar processo:** Arquivar todo o processo no RIS para posterior consulta.

É de referir que estes procedimentos representam o grosso do trabalho dos médicos mas existirão certamente situações que não foram contempladas. Por exemplo uma delas poderá ser a visualização e especificação de parâmetros de imagens geradas por técnicas 3D.

Neste trabalho os pontos 3, 4 e 5 serão mais focados, embora os restantes também sejam também contemplados, mas não explorados tão exaustivamente. Isto porque o trabalho despendido seria extremamente grande e demasiado vasto.

Capítulo 3

Visualizadores e Sistemas de Visualização de Imagens Médicas

Actualmente os médicos de áreas relacionadas com a imagiologia recorrem a visualizadores ou a sistemas de visualização de imagens médicas para poderem navegar e analisar as mesmas. No contexto deste trabalho não entramos em pormenores acerca das diferenças entre um visualizador e um sistema de visualização de imagens médicas, contudo é importante diferenciá-los. Um visualizador de imagens médicas é uma aplicação de *software* que permite visualizar dados médicos e.g. num ecrã de um PC convencional. Quanto ao termo sistema de visualização, este é muito mais abrangente: trata-se efectivamente de um sistema que implementa o processo de diagnóstico por imagem médica, logo constituído por blocos componentes que interagem entre si para formar um todo. Estes componentes tanto podem ser de *software* como de *hardware*. Por exemplo um sistema de visualização de imagens médicas pode possuir n-monitores e n-visualizadores, todos componentes do mesmo sistema.

Existem inúmeras soluções disponíveis, no entanto constata-se que a generalidade destes programas de visualização de imagem médica continua a apresentar o mesmo esquema para análise das imagens, consistindo numa alteração sequencial de uma ou um conjunto de imagens visualizadas.

3.1 Soluções Existentes

De entre as soluções existentes, seleccionamos um conjunto para análise tendo em atenção fundamentalmente dois factores: serem amplamente usadas no meio médico/académico e apresentarem ferramentas/funcionalidades relevantes. Os sistemas de visualização e visualizadores seleccionados são apresentados nas secções seguintes.

3.1.1 Merge eFilm™

A empresa Merge eFilm™ [WEB-EFILM'05] foi fundada em 1987 e nasceu da união de três empresas líderes no desenvolvimento de *software* para radiologia: a Merge Technologies Incorporated, a eFilm Medical, Inc. e a RIS Logic, Inc.

Actualmente é uma das empresas líderes de mercado com mais de 30.000 sistemas vendidos por todo o mundo.

A Merge eFilm™, possui vários componentes de sistemas que podem ser encontrados no seu *website*. Aquele que nos interessa directamente é o eFilm Worstation™, pois este é o que possibilita a visualização dos dados. O FUSION eFilm™, estende as funcionalidades do eFilm Worstation™, mas devido ao facto dos componentes FUSION serem utilizados em ambientes clínicos para a fusão dos vários módulos existentes e estes não estarem disponíveis para testes, não foi possível analisar estas funcionalidades.

Devido à aquisição da empresa AccuImage, por parte da Merge eFilm™, existe uma nova solução chamada AccuImage que será integrada nos produtos existentes para aumentar as suas funcionalidades de visualização avançada.

A versão analisada neste trabalho é a eFilm Worstation™ 2.0.1.

3.1.2 DicomWorks

O DicomWorks [WEB-DICOMWORKS'05] é um *software* livre desenvolvido por Philippe Puech e Loïc Boussel. Começou a ser desenvolvido em Setembro de 2000, com o objectivo de criar sequências AVI (*Audio Video Interleave*) partindo de séries DICOM. Contudo, com o desenrolar do projecto este foi incorporando mais funcionalidades e actualmente é um visualizador DICOM completo.

A versão analisada neste trabalho é a DicomWorks 1.3.5.

3.1.3 Rubo Medical

A Rubo Medical Imaging [WEB-RUBOMEDICAL'05] iniciou o desenvolvimento de visualizadores em 1991, em cooperação com a Erasmus University Rotterdam, na Holanda, possuindo actualmente um dos visualizadores mais completos do mercado.

A avaliação dos seus produtos é feita em vários hospitais e laboratórios por todo o mundo, os principais são: a Erasmus University Rotterdam, University of Groningen, Instituto do coração de São Paulo e o Hospital de Vila Nova de Gaia.

A versão analisada neste trabalho é a Rubo Dicom Viewer 2.0.

3.1.4 Osiris

Desenvolvido pela Unidade de Imagem Digital do Serviço de Informação Médica do Hospital de Genève, na Suíça, este é também um visualizador não comercial. É de referir que existe uma versão para o sistema operativo MacOSX com o nome Osirix.

A versão analisada neste trabalho é a Osiris 4.19 [WEB-OSIRIS'05], para o sistema operativo Windows XP.

3.1.5 ImageJ

O ImageJ [WEB-IMAGEJ'05] é uma aplicação de processamento e análise de imagem desenvolvida utilizando a linguagem de programação Java. O seu autor é Wayne Rasband, do National Institute of Mental Health, Bethesda, Maryland, USA.

A sua arquitectura foi desenvolvida para permitir extensões, através de *plugins* escritos em Java. Esta característica é sem dúvida a mais valia desta aplicação, porque existem actualmente inúmeros *plugins* criados, como por exemplo *plugins* que permitem a visualização de imagens DICOM 2D e 3D. Este visualizador e os seus *plugins* são livres.

A versão analisada é a ImageJ 1.33u.

3.1.6 MRICro

O MRICro [WEB-MRICRO'05] é mais um visualizador de imagem médica disponível, gratuitamente desenvolvido por Chris Rorden da University of Nottingham. Este visualizador explora principalmente as técnicas de reconstrução 3D, *Volume e Surface Rendering*.

A versão analisada neste trabalho é a MRICro 1.39.

3.1.7 Julius

O projecto Julius [WEB-JULIUS'05] teve início no final de 1999, sendo a primeira tarefa do recentemente fundado Surgical Systems Laboratory, do centro de pesquisa Caesar. Este é um *software development framework* que tem licenças comerciais e não comerciais. Possui um visualizador chamado JuliusLight que permite utilizar e testar o *framework*.

A versão utilizada é a JuliusLight 1.0.0.

3.2 Análise das Soluções Existentes

As soluções existentes descritas anteriormente possuem algumas características importantes que são apresentadas nas secções seguintes.

3.2.1 Ferramentas de Medição

Os especialistas recorrem à utilização de ferramentas de medição para avaliar diversas patologias. Os visualizadores em que foi constatado o suporte destas ferramentas são:

- Merge eFilm™
- DicomWorks
- Rubo Medical
- Osiris
- Julius

Alguns dos visualizadores podem, eventualmente, suportar estas funcionalidades, como é o caso do ImageJ (através de *plugins*): no entanto, na lista acima, são apenas apresentados aqueles em que a utilização é bem definida e intuitiva.

3.2.2 Ferramentas de zoom

Todos os visualizadores possuem alguma forma de *zoom*. Por exemplo em alguns casos basta aumentar a janela que contem a imagem. Outros aumentam uma área quadrada definida pela posição corrente do cursor do rato. Existem ainda outras formas de *zoom*, mas o mais interessante é que alguns dos visualizadores utilizam não apenas uma, mas uma combinação dos vários tipos.

3.2.3 Ferramentas de controlo do centro e largura da janela

O centro e a largura da janela são parâmetros utilizados para controlar facilmente o brilho e o contraste das imagens. Todos os visualizadores possuem alguma forma de controlo destes parâmetros, quer seja através de *scrollbars*, movimento do rato e botão pressionado, etc.

3.2.4 Estrutura hierárquica de informação do paciente

A norma DICOM define uma estrutura hierárquica em árvore que contempla vários níveis (figura 1). Os visualizadores que apresentam a informação sobre o paciente de uma forma hierarquizada são:

- Merge eFilm™
- DicomWorks
- Rubo Medical
- Osiris

3.2.5 Processamento de imagem

Alguns visualizadores possuem ferramentas sofisticadas para processamento de imagem, sendo uma das mais utilizadas a segmentação para identificação de órgãos. A título de exemplo: em [MCINERNEY&TERZOPOULOS'99] e [FAN et al.'02] podem ser encontrados exemplos da utilização destas técnicas de segmentação.

As duas aplicações mais especializadas no processamento de imagem são: o ImageJ e o Osiris que possuem um conjunto muito vasto de algoritmos disponíveis, utilizados em situações particulares e de difícil análise.

3.2.6 Possibilidade de visualizar várias imagens da mesma série

Muitos dos visualizadores existentes não possibilitam a visualização de várias imagens de uma série ao mesmo tempo, e.g. é utilizada uma *scrollbar* para mudar entre as imagens da série, como apresentado na figura 6. Contudo, existem visualizadores como o eFilm™ e o DicomWorks que possibilitam a visualização de diversas imagens da mesma série ao mesmo tempo. Esta funcionalidade é importante pois permite uma comparação imediata das imagens e não indirectamente através da *scrollbar*. Tornando-se ainda mais importante, quando é possível colocar imagens de séries diferentes no mesmo ecrã.

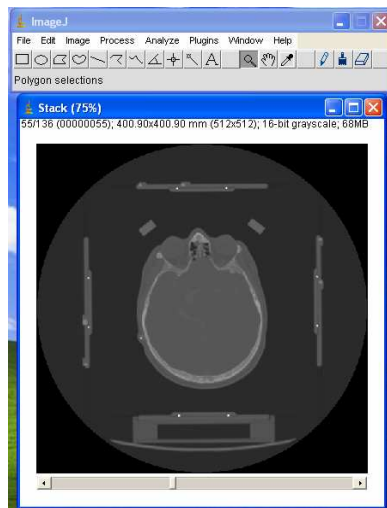


Figura 6: *Scrollbar* horizontal para navegação pelas imagens de uma série

O visualizador que melhor explora a disposição de imagens, i.e. o *layout* ou protocolo de afixação, é o eFilm™, permitindo subdividir o ecrã em grelhas. Possui dois níveis: o nível das séries e o nível das imagens. As imagens encontram-se no nível inferior, ou seja, dentro das células das séries. O número de linhas e colunas das séries e imagens é especificado pelo utilizador, como apresentado na figura 7.

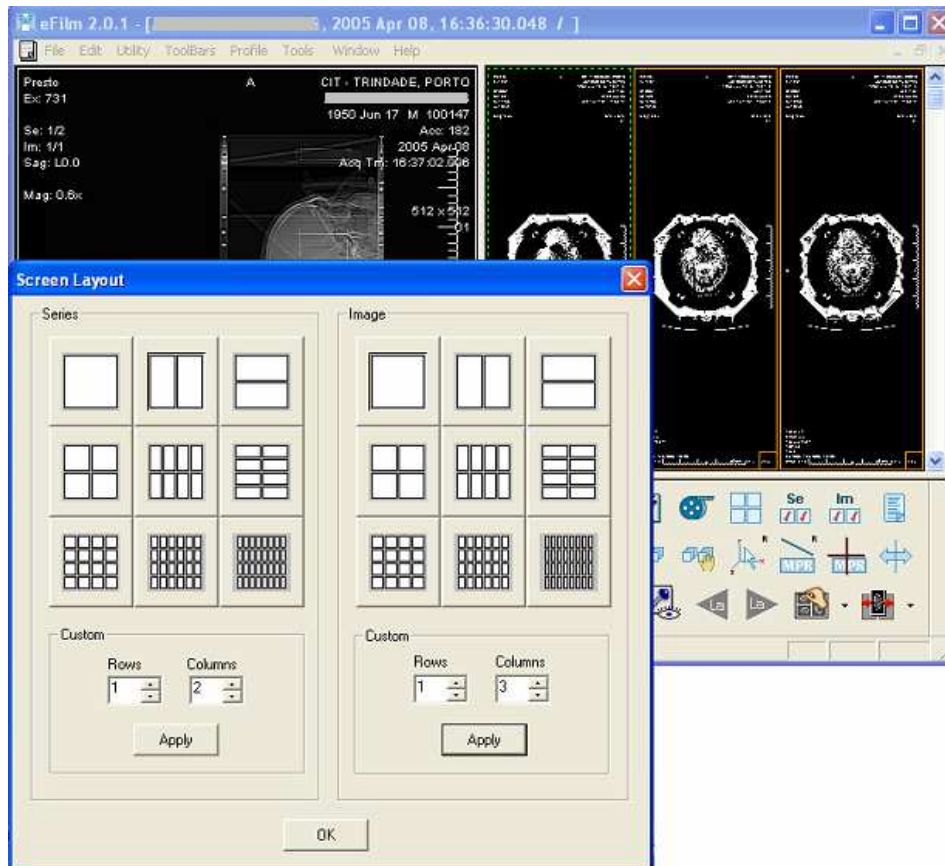


Figura 7: Especificação do *Layout* utilizando o eFilm™

3.2.7 Volume e Surface Rendering

A maioria dos visualizadores apresentados possui alguma forma de *Volume* ou *Surface Rendering*. Porém, os dois visualizadores especializados nesta área são o MRIcro e o Julius. É de referir que é necessário algum cuidado na utilização destas aplicações, isto porque os visualizadores carregam para a memória todas as imagens de uma série ou exame e também os dados necessários para a reconstrução (por exemplo gradientes), o que facilmente esgota os recursos computacionais existentes, devido ao enorme volume de dados. Isto faz com que a exploração e análise de dados recorrendo a estas técnicas 3D seja feita muito cuidadosamente. É necessário ainda referir que a parametrização das variáveis para controlar a visualização do volume, é bastante mais complexa, em comparação com a parametrização das variáveis (e.g. centro e largura da janela), para visualizar zonas de interesse nas imagens 2D. Em *Volume Rendering* as variáveis produzem uma função de transferência, ou seja, a transformação das densidades do material (os valores escalares originais) em cor e opacidade que irão ser utilizados para produzir a imagem final. Para a obtenção dos valores de cor recorre-se geralmente a uma *lookup table*. Um exemplo da parametrização de uma *lookup table* encontra-se na figura 8.

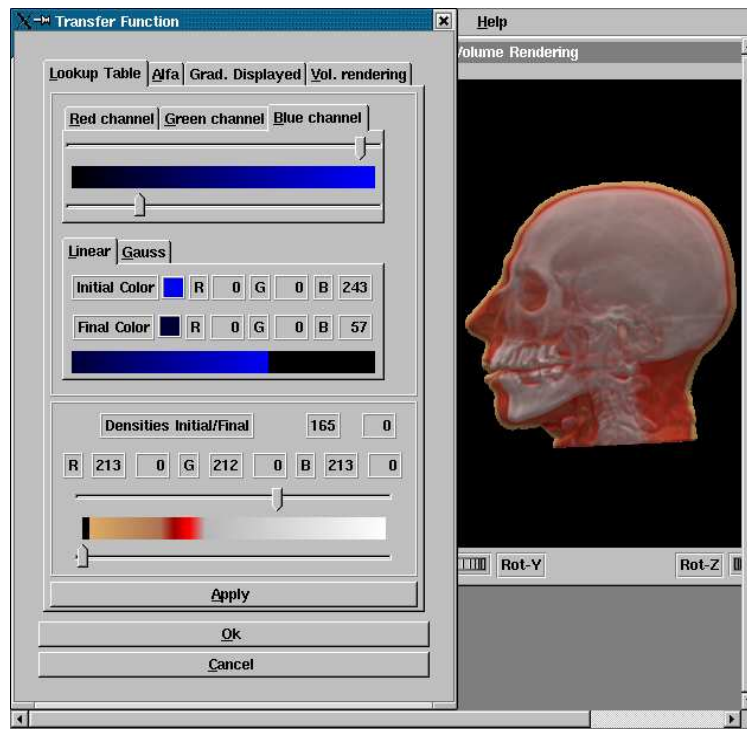


Figura 8: Lookup Table

O visualizador apresentado acima foi desenvolvido para o projecto de conclusão da licenciatura do autor deste trabalho, intitulado *Realtime Direct Volume Rendering* [MARREIROS'02].

Como se pode observar ainda na figura 8, controlando a opacidade atribuída aos materiais podemos produzir zonas semitransparentes. Possuindo estas funcionalidades é possível simular *Surface Rendering*, para tal basta atribuir a todas as densidades uma opacidade máxima, como no caso apresentado na figura 9.

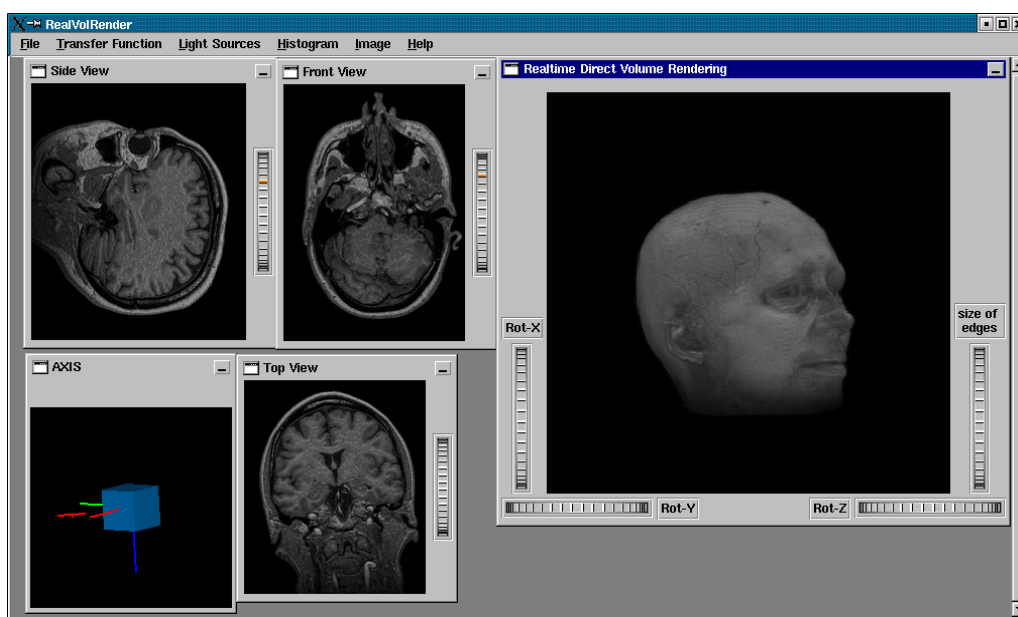


Figura 9: Surface Rendering

Estas manipulações produzem um aumento do tempo despendido na análise dos dados, o que em muitos dos casos não é aceitável. Para tentar auxiliar o processo de manipulação da função de transferência existem, e.g. duas abordagens: gerar um conjunto elevado de imagens que representam diversas funções transferência e a utilização de interfaces que facilitam a parametrização das funções transferência. Existem ainda classificações alternativas, como as descritas em [PFISTER et al.'01] onde são apresentadas exemplos das formas consideradas para obtenção das funções de transferência. Considerando a classificação inicial, pode ser encontrado em [WEB-VOLDG'05] um exemplo claro da primeira abordagem, ou seja: automaticamente a aplicação cria um conjunto elevado de imagens que representam diversas funções de transferência e assim o utilizador apenas terá de escolher qual a que melhor se adequa às suas necessidades. Quanto à segunda abordagem, existe um conjunto vasto de interfaces desenvolvidos para o auxílio da parametrização das funções transferência, de entre este conjunto destacam-se dois trabalhos relevantes [KINDLMAN&DURKIN'98] e [TORY et al.'05].

Nos últimos anos têm vindo a ser investigadas aplicações de *Volume Rendering* no campo da medicina recorrendo a técnicas de realidade virtual, nomeadamente estereoscópicas. Através destas técnicas os médicos podem visualizar as imagens com profundidade. Este factor é bastante interessante, por exemplo quando se quer explicar a um paciente onde se localiza um determinada patologia no seu corpo [TSCHIRLEY et al.'03]. Mas existem inúmeras aplicações onde estas técnicas podem ser empregues. Para além dos dispositivos tradicionais que possibilitam estereoscopia, existem actualmente outros dispositivos que podem ser utilizados, como por exemplo os monitores 3D e os *VirtualShowcases* [BIMBER et al.'01]. Os monitores 3D também conhecidos por monitores *Autostereo* não necessitam dos óculos (passivos ou activos) para que a visualização seja possível. Em [WEB-STEREO3D'05] encontram-se apresentadas um conjunto vasto de soluções existentes. Quanto aos *VirtualShowcases*, estes permitem que um objecto seja visualizado no interior de um *Showcase*, podendo o utilizador livremente visualizar o objecto se possuir um sistema de detecção da sua posição, utilizado para que se possa calcular correctamente as imagens apresentadas [BIMBER et al.'02], [MARREIROS et al.'05a], [MARREIROS et al.'05b].

Contudo a utilização destas técnicas 3D ainda se encontra no início e prevê-se que, com os avanços tecnológicos futuros elas sejam amplamente utilizadas, pois permitem estudar e analisar o volume completo de dados, directamente.

3.2.8 Multidisplay

Actualmente os sistemas operativos possuem funcionalidades de *multidisplay*. É vulgar encontrar um PC convencional ligado a dois monitores utilizando por exemplo o sistema

operativo Windows XP. Mas nem todos os visualizadores contemplam da forma mais apropriada estas características. Dois visualizadores que possuem directamente algumas funcionalidades de *multidisplay* são o Merge eFilmTM e o Rubo Medical. Ambos permitem especificar se a visualização é realizada apenas considerando um monitor ou vários. Contudo não é conhecida uma solução realmente escalável, i.e., um sistema onde ligando mais PCs se possa utilizar os recursos (monitor, etc.) dos mesmos.

Capítulo 4

Sistema Multiagente

O Sistema Multiagente, desenvolvido no contexto deste trabalho, pertence à área da Inteligência Artificial Distribuída. A escolha por este tipo de sistemas deve-se à sua versatilidade e escalabilidade entre outras características relevantes, nomeadamente, permitiu uma distribuição eficiente das tarefas fundamentais para o sucesso geral do sistema. Para compreendermos como tal foi conseguido, é necessário introduzir conceitos referentes aos Sistemas Multiagente, antes de descrevermos e especificarmos em detalhe todos os componentes e objectivos do sistema.

4.1 Sistemas Multiagente

Os Sistemas Multiagente (SM) formam uma sub-área da Inteligência Artificial Distribuída. São constituídos por um universo de Agentes Autónomos que possuem um conjunto de propriedades associadas. Existe uma quantidade enorme de definições para o termo Agente, em [FRANKLIN&GRASSER'96] encontramos um sumário das definições utilizadas e ainda uma classificação de agente desenvolvida no contexto do trabalho referido. Para além da definição de agente, é também necessário conhecermos como estes interagem entre si e/ou com o ambiente que os rodeia, e de que forma se pode especificar formalmente o sistema multiagente.

4.1.1 Agentes

A definição de agente sempre levantou inúmeros debates na comunidade da Inteligência Artificial [FRANKLIN&GRASSER'96]. Não abordaremos todas as características debatidas nas várias definições existentes, o estudo será restringido às características e propriedades consideradas relevantes no contexto deste trabalho.

Em [MOULIN&CHAIB-DRAA'96] são distinguidos os dois tipos fundamentais de agentes:

- **agentes artificiais**, onde se incluem os **agentes de *software***,
- **agentes humanos**, e.g., usuários.

As classificações, que serão aplicadas posteriormente, referem-se apenas aos agentes do tipo artificial.

Uma das classificações mais frequente para agente, é encontrada em [SICHMAN et al.'92], baseando-se nas diversas capacidades dos agentes para a resolução de problemas. A classificação proposta classifica-os em: agentes reactivos e agentes cognitivos.

4.1.1.1 Agentes Reactivos

Os agentes reactivos, comportam-se segundo o modelo Estímulo-Resposta (Par Acção-Reacção). Este tipo de agente, não possui, memória sobre as acções realizadas no passado e nem efectua previsões para acções futuras. As principais características destes agentes e dos sistemas multiagentes reactivos, encontram-se sumariadas em [FERBER&GASSER'91], e são apresentadas de seguida:

- não existe representação explícita do conhecimento: o conhecimento dos agentes é implícito (segundo as suas regras de comportamento);
- não existe uma representação do ambiente: o comportamento de cada agente é baseado no que ele percebe (estímulo) a cada instante;
- não possui memória: os agentes reactivos não mantêm nenhum tipo de histórico das suas acções, ou seja, o resultado de uma determinada acção passada não influencia directamente na tomada de decisão de uma acção futura;
- organização etológica/biológica: a forma de organização dos SM reactivos é similar à observada por animais que vivem em grandes comunidades;
- grande número de membros: em geral, os SM reactivos possuem um grande número de agentes, com populações que podem chegar à ordem de milhares de membros.

Em suma, os agentes reactivos são muito simples, não possuem representação do seu ambiente nem memória, limitando-se a reagir de acordo com a sua percepção do ambiente. Contudo como estes sistemas possuem uma grande quantidade de agentes, a realizarem tarefas simples, o resultado do conjunto global permite que possam ser realizadas tarefas bastante complexas. Uma analogia comum é realizada com as colónias de formigas: uma formiga sozinha não é capaz de realizar tarefas muito inteligentes, mas uma colónia sim.

4.1.1.2 Agentes Cognitivos

Os agentes cognitivos, são baseados em organizações sociais humanas como: grupos, hierarquias e mercados. Segundo [FERBER&GASSER'91], as principais características dos Sistemas Multiagentes Cognitivos são as seguintes:

- representação explícita do ambiente e dos outros agentes da sociedade;
- podem manter um histórico das interacções e acções passadas e, devido a esta memória são capazes de planear as suas acções futuras;

- os seus sistemas de percepção (que permitem examinar o ambiente) e o de comunicação (que permite a troca de mensagens entre agentes) são distintos. A comunicação entre agentes é realizada de modo directo, através do envio e recepção de mensagens.
- o seu mecanismo de controlo é deliberativo. Os agentes cognitivos raciocinam e decidem em conjunto sobre quais as acções que devem ser executadas, que planos a seguir e que objectivos devem ser alcançados.
- Os modelos de organização dos SM cognitivos são modelos sociológicos, como as organizações humanas.

Para formar um grupo social unido, é necessário que algumas questões sejam tratadas [ROCHA et al.'03]:

- Organização: como os agentes interagem entre si, e qual o tipo de organização que eles adoptam;
- Cooperação: quando um agente não estiver capacitado para realizar sozinho uma tarefa pessoal, ele deve cooperar com outros agentes. Esta cooperação deve ainda ocorrer quando outros agentes poderem executar mais eficientemente a mesma tarefa;
- Negociação: como é realizada a negociação entre os agentes, dividindo a execução das tarefas, de maneira que seja mais organizada e fazendo uso das capacidades e conhecimentos dos agentes;
- Comunicação: como os agentes irão comunicar e qual o protocolo de comunicação que será utilizado na interacção entre esses agentes;

4.1.2 Interacções nas sociedades de agentes

De acordo com [OLIVEIRA'96] existem três critérios para classificar uma sociedade de agentes:

- Quanto ao tipo de agente:
 - Sociedades homogéneas: os agentes são todos do mesmo tipo, i.e., possuem arquitecturas idênticas;
 - Sociedades heterogéneas: existem agentes de diversos tipos na sociedade.
- Quanto à migração dos agentes:
 - Sociedades fechadas: existe um número fixo de agentes presentes na sociedade;
 - Sociedades abertas: o número de agentes nesta sociedade pode variar, pois podem entrar novos agentes ou sair agentes da sociedade.

- Quanto à presença de regras de comportamento:
 - Sociedades baseadas em leis: existem regras que determinam o comportamento dos agentes;
 - Sociedades sem leis: quando não há regras para reger os agentes da sociedade.

Para que seja possível a correcta cooperação entre os agentes, é necessário definir uma arquitectura que possibilite a interacção entre os agentes. Nestas interacções poderão ser trocadas entre os agentes: conhecimento, objectivos e planos. Para tal é necessário que eles possuam uma forma de comunicação, que pode ser de dois tipos:

- Comunicação directa: os agentes conhecem-se e, por isso, trocam informações directamente entre si;
- Comunicação indirecta: os agentes não se conhecem, devido a este facto os agentes comunicam através de uma memória partilhada, onde depositam e retiram mensagens.

4.1.2.1 Arquitectura do Quadro Negro

Quando os agentes não se conhecem, é necessário, que exista uma estrutura de dados partilhada (memória partilhada) para permitir a interacção. Uma abordagem para a resolução do problema de comunicação é o Quadro Negro (*blackboard*).

O Quadro Negro é uma estrutura de dados que os agentes recorrem para: deixar mensagens a outros agentes e ler mensagens destinadas a si próprios provenientes de outros agentes. O que implica que, os agentes têm de verificar em determinados intervalos de tempo, se existe informação destinada a eles. A figura 10 apresenta a estrutura do Quadro Negro.

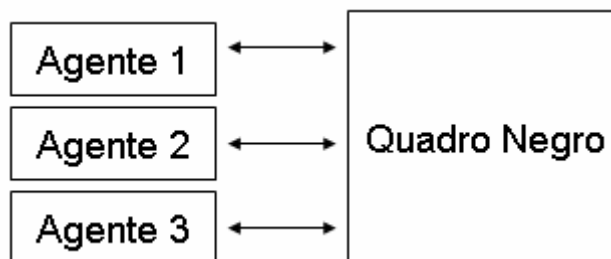


Figura 10: Estrutura do Quadro Negro

As estruturas de Quadro Negro são anteriores aos sistemas multiagente. Um dos primeiros sistemas a utilizar essa abordagem foi o HEARSAY II [OLIVEIRA'96], um sistema para a interpretação da fala.

Embora esta abordagem seja simples de implementar e eficiente, um dos problemas que ocorre na estrutura de Quadro Negro é a sincronização e resolução de conflitos entre os agentes, devido à centralização da informação de controlo.

4.1.2.2 Troca de Mensagens

Nas sociedades onde os agentes se conhecem a comunicação pode ser directa, ou seja, os agentes trocam mensagens entre si directamente, para atingirem um objectivo global. Implicando que os agentes necessitam de possuir representações dos outros agentes, tais como: capacidades, objectivos, conhecimento e crenças.

Segundo [DEMAZEAU&MÜLLER'90], existem basicamente três tipos de interacções entre agentes. Neste caso, o tipo de informação que será partilhado, (conhecimento, possibilidades ou escolhas) irá determinar o tipo de interacção. Os tipos de interacção são:

- Interacção forte: quando os agentes compartilham conhecimentos, possibilidades e escolhas. Neste tipo de comunicação ocorrem protocolos de comunicação sofisticados para os agentes, tais como: informar, requisitar ou convencer;
- Interacção média: quando os agentes compartilham apenas conhecimentos e possibilidades. Este tipo de interacção ocorre quando os agentes desejam apenas saber o que os outros agentes pretendem fazer. Desta forma os agentes irão conhecer os planos para execução de tarefas dos outros agentes, até que um plano comum seja encontrado;
- Interacção fraca: quando os agentes compartilham apenas conhecimentos. Esta interacção ocorre através de trocas de mensagens que serão ocasionadas pela percepção de alterações no ambiente.

4.1.3 Especificação de sistemas multiagente

Existem diversas formas de especificar formalmente sistemas multiagente, neste trabalho foi adoptado um formalismo desenvolvido para a especificação, modelação e raciocínio de um SM que utiliza a noção de contexto e incorpora algumas facilidades da programação orientada ao objecto, nomeadamente: a abstracção, o encapsulamento, a modularidade e a hierarquia [CAVEDON&TILHAR'95], [NEVES&MACHADO'97], [SANTOS'99] e [ALVES'02]. Um contexto pode ser interpretado (num determinado nível de abstracção) como um ambiente, uma hierarquia de componentes de *software* com herança, uma instância de um objecto, uma linha de raciocínio, um subsistema, um agente ou uma comunidade de agentes.

Através deste formalismo é possível de uma forma flexível e modular, especificar:

- Os componentes do sistema ou agentes;
- O processo de socialização ou formas possíveis de agregação e cooperação;
- O procedimento de coordenação dos agentes;
- O comportamento global do sistema.

Este formalismo permite a especificação de cada uma das entidades que compõe o sistema, uma vez que cada entidade apenas reporta ao seu próprio universo, assim como as interações entre si. Nesta óptica, cada agente é modelado num contexto em separado, com a sua própria lógica, o que apresenta vantagens competitivas, sendo desejáveis nestas entidades, entre outras, algumas das características enunciadas:

- Cada agente é uma entidade tão simples quanto possível, de tal forma que um agente apenas pode ser relacionado com as proposições no mundo ou universo de discurso com o qual está directamente relacionado;
- Agentes diferentes podem ser modelados como tendo interesses distintos, ou pelo menos perspectivas diferentes do mundo dentro do seu universo de discurso;
- O desempenho de um agente pode ser maximizado, desde que lhe seja limitado o acesso a conhecimento, na medida em que se está em presença de conhecimento que se aplica à resolução de um certo tipo de problemas.

Consequentemente, os agentes utilizados poderão ser dos dois tipos estudados anteriormente: agentes reactivos e agentes cognitivos ou um tipo de agente que possui uma combinação das propriedades de ambos. Como exemplo, considere-se que possuímos um agente reactivo, este apenas necessita de sentir o ambiente, ou seja, neste caso o ambiente actua sobre o contexto do agente forçando o agente a reagir de acordo com as alterações do mesmo, mas se o agente possui conhecimento sobre outros agentes, ou seja, o seu contexto possui uma representação do sistema ou subsistema, este agente aproxima-se mais do tipo de agente cognitivo. Isto significa que a classificação de agente não é rígida podendo o agente apresentar características de ambos os tipos.

Segundo a abordagem utilizada, um SM ou um Sistema de Processamento Distribuído, pode ser formalmente definido através de:

- Um contexto que denota uma teoria em particular;
- Um conjunto de regras-ponte que funcionam como o interface entre os agentes e entre sistemas de agentes (comunidades);
- Um conjunto não vazio de subsistemas, correspondendo cada um deles, por sua vez, a um multi-sistema com o seu próprio contexto.

As regras-ponte dão corpo, de uma forma explícita, à interacção e coordenação entre dois subsistemas. As restrições impostas por estas regras reflectem a esperança de que qualquer procedimento seja efectuado (com privacidade) dentro dos componentes do sistema (racionalidade emergente). Os componentes de um sistema não são visíveis fora desse sistema.

As regras-ponte podem assumir a forma de:

- Regras subordinantes na forma $\frac{ocorre(e,i)}{C_k : ocorre(e',i')}$, denotando que se no sistema ocorre o evento e no instante i , então no contexto C_k ocorrerá o evento e' no instante i' ;
- Regras subordinadas na forma $\frac{C_k : ocorre(e,i)}{ocorre(e',i')}$, denotando que se num determinado contexto C_k ocorre o evento e no instante i , então no sistema ocorrerá o evento e' no instante i' ;
- Regras ordinárias na forma $\frac{C_k : ocorre(e,i)}{C_l : ocorre(e',i')}$, denotando que se num determinado contexto C_k ocorre o evento e no instante i , então no contexto C_l ocorrerá o evento e' no instante i' .

4.2 Especificação do Sistema

Utilizando as ferramentas de especificação de sistemas multiagente, iremos especificar o sistema desenvolvido no contexto deste trabalho. No entanto a especificação do sistema não se restringe apenas à formalização lógica apresentada anteriormente, nas secções seguintes serão apresentadas todas as ferramentas conceptuais utilizadas na especificação do sistema.

4.2.1 Objectivos Globais

Quando se especifica um sistema deste tipo, é necessário definir criteriosamente quais os objectivos globais, para que possa especificar, correctamente, quais os componentes necessários para levar a bom termo os problemas a resolver. No nosso caso o sistema tem por objectivos:

- Efectuar o *parsing* dos ficheiros DICOM, para que o sistema possa interpretar a informação contida nos mesmos;
- Visualizar a informação médica em vários terminais de visualização (*displays*);
- Controlo, através de uma estação de controlo, do conteúdo visível nos terminais de visualização do sistema.

Existe ainda a necessidade de enviar a informação médica para as várias máquinas (terminais), quando a informação é carregada na estação de controlo. Como o sistema é *multidisplay* é necessário que as imagens se encontrem disponíveis nos vários terminais de visualização e que estas possam ser alteradas (posicionamento, cor, etc.) utilizando a estação de controlo.

4.2.2 Arquitectura

Para alcançar os objectivos propostos foi desenvolvida a arquitectura apresentada na figura 11, com base nas suas entidades constituintes:

- O Agente de Preparação de Dados (*apd*) – Encarregado de verificar quais os exames existentes na *worklist* do RIS e preparar (*parsing*) esses exames para poderem ser utilizados pelo sistema;
- O Agente da Estação de Controlo (*aec*) – Encarregado da interface entre os utilizadores e o sistema. É ainda responsável por verificar os exames da *worklist* e enviar a informação para os terminais de visualização;
- Os Agentes dos Terminais de Visualização (*atv_i*) – Encarregados da visualização das imagens nos vários terminais de visualização do sistema.

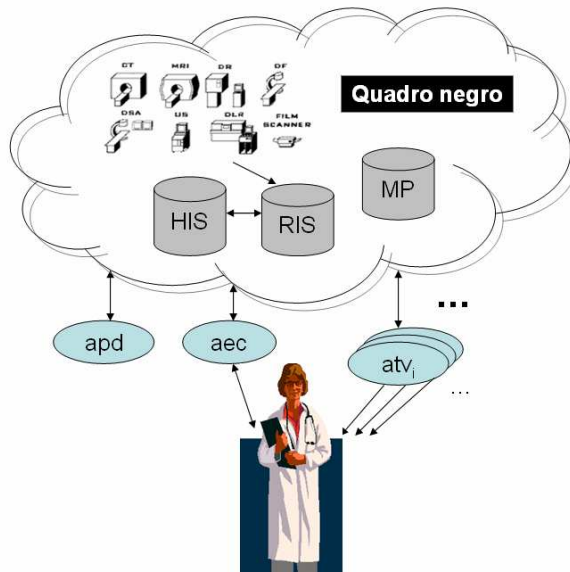


Figura 11: Arquitectura do sistema multiagente

Formalmente o sistema dá forma a um multi-sistema $\mathcal{E} = \langle \mathcal{C}s, \Delta s, apd, aec, atv_1, \dots, atv_i, qenv \rangle$, com $i \geq 1$, em que:

- **$\mathcal{C}s$** – denota o contexto em que o sistema se insere e opera;
- **Δs** – denota o conjunto de regras-ponte, que definem o processo de interacção entre os componentes do sistema. Definem a geometria do sistema, dando uma medida do significado e da interacção possível entre as partes constituintes. Estabelecem o modo

como os eventos que ocorrem ao nível dos subsistemas ou agentes são coordenados, e expressam o grau de paralelismo presente no sistema;

- ***apd*** – denota o agente de preparação de dados;
- ***aec*** – denota o agente estação de controlo;
- ***atv_i*** – denota um agente terminal de visualização ($i=1...n$), em que n é um parâmetro do sistema; e
- ***qenv*** – denota o ambiente ou universo de discurso.

As regras-ponte do sistema são apresentadas na tabela 1.

Tabela 1: Regras-ponte do sistema multiagente

Regra-ponte	Descrição
$\frac{ocorre(exame_dicom)}{C_{apd} : ocorre(interpreta_dicom)}$	O sistema possui um novo exame. O agente de preparação de dados irá efectuar a interpretação do exame DICOM e salvar os ficheiros resultantes no directório dos ficheiros preparados.
$\frac{C_{apd} : ocorre(interpreta_dicom)}{C_{aec} : ocorre(actualiza_worklist)}$	Quando o agente de preparação de dados termina a interpretação do exame, ele irá escrever o caminho do novo exame num ficheiro no disco rígido. Este ficheiro é verificado constantemente pelo agente estação de controlo, desta forma podemos actualizar a sua <i>worklist</i> .
$\frac{C_{atv_i} : ocorre(activa)}{C_{aec} : ocorre(actualiza_vterminais)}$	Se um agente terminal de visualização é activado, a lista de terminais de visualização é actualizada na estação de controlo.
$\frac{C_{aec} : ocorre(activa)}{C_{aec} : ocorre(inicia_vterminais)}$	Quando a estação de controlo é activada a lista de terminais de visualização é iniciada.
$\frac{C_{aec} : ocorre(inicia_vterminais) \vee C_{aec} : ocorre(actualiza_vterminais)}{C_{aec} : ocorre(recarregamento_series)}$	Quando a lista de terminais de visualização é iniciada ou actualizada é necessário efectuar um recarregamento (caso alguma das séries se encontre carregada), para que todos os terminais possuam a mesma informação.
$\frac{C_{aec} : ocorre(alteração_layout)}{C_{atv_i} : ocorre(alteração_layout)}$	Quando o <i>layout</i> é alterado na estação de controlo esta alteração tem de ser reflectida nos terminais de visualização. O <i>layout</i> é alterado nas seguintes condições: <ul style="list-style-type: none"> - quando o utilizador altera o posicionamento das imagens; - quando é efectuado o carregamento ou recarregando de séries; - quando o <i>layout</i> das imagens é alterado; - quando o <i>layout</i> das séries é alterado.

$\frac{C_{aec} : ocorre(recarregamento_series) \vee C_{aec} : ocorre(carregamento_series)}{C_{atv_i} : ocorre(carregamento_series) \wedge C_{aec} : ocorre(alteração_atributosimagens) \wedge C_{aec} : ocorre(alteração_layout)}$	<p>Quando uma série é carregada ou recarregada na estação de controlo, esta terá de ser também carregada nos terminais de visualização. E os atributos das imagens das séries e o <i>layout</i>, necessitam de ser alterados de acordo com a nova informação.</p>
$\frac{C_{aec} : ocorre(alteração_atributosimagens)}{C_{atv_i} : ocorre(alteração_atributosimagens)}$	<p>Quando os atributos das imagens de uma série são alterados na estação de controlo eles também têm de ser alterados nos terminais de visualização. Os atributos são alterados quando:</p> <ul style="list-style-type: none"> - o utilizador altera o centro e largura da janela da imagem; - o utilizador efectua uma rotação ou <i>flip</i>; - o utilizador efectua uma inversão de cor. <p>Estas alterações não são efectuadas em tempo real, porque seria um processo muito dispendioso enviar a informação e actualizar as imagens ao mesmo tempo. Por esta razão as imagens são actualizadas apenas quando o utilizador sai do nível de visualização superior (onde ele/ela pode analisar uma única imagem e alterar os seus parâmetros em tempo real).</p>
$\frac{C_{aec} : ocorre(descarregamento_series)}{C_{atv_i} : ocorre(descarregamento_series)}$	<p>Quando uma série é descarregada na estação de controlo, esta também terá de ser descarregada nos terminais de visualização.</p>
$\frac{C_{aec} : ocorre(exame_analizado)}{C_{aec} : ocorre(actualizar_woklist) \wedge C_{aec} : ocorre(descarregamento_series)}$	<p>Quando um exame terminou de ser analisado, este será descartado da <i>worklist</i>. O exame preparado será eliminado do disco rígido e a sua referência excluída do ficheiro da <i>worklist</i>. Também a(s) série(s) correspondente(s) será(ão) descarregadas.</p>
$\frac{C_{aec} : ocorre(falha_comunicação_rede)}{C_{aec} : ocorre(desactiva)}$	<p>Quando é detectada uma falha de comunicação, a estação de controlo é desactivada e pedido ao utilizador para verificar o estado da sua rede.</p>
$\frac{C_{aec} : ocorre(desactiva)}{ocorre(desactiva)}$	<p>Quando a estação de controlo é desactivada esta desactiva também os outros agentes do sistema.</p>

4.2.3 Estrutura Lógica

A estrutura lógica define o conjunto de funcionalidades inerentes aos agentes do sistema. Isto inclui ou passa pela caracterização formal dos agentes do sistema e do meio envolvente, que será efectuada nas secções seguintes.

4.2.3.1 O Agente de Preparação de Dados, *apd*

O agente de preparação de dados é encarregado de verificar quais os exames existentes na *worklist* do RIS e preparar (*parsing*) esses exames para poderem ser utilizados pelo sistema. Por cada imagem DICOM processada serão gerados três ficheiros: o primeiro contém as *Tags* DICOM, o segundo os valores brutos (*raw data*) das imagens e o terceiro um conjunto de atributos comuns às imagens de cada série (exemplo: número de imagens, dimensão das imagens, etc.). Do último tipo existe apenas um ficheiro por série, no entanto o seu conteúdo vai sendo actualizado à medida que o *parsing* é efectuado, ao longo das imagens de uma série.

A informação do caminho onde gravar os exames preparados é mantida num ficheiro no disco rígido. Este ficheiro pertence ao ambiente e o agente em causa verifica e utiliza o seu conteúdo quando necessário. O utilizador poderá alterar o caminho onde pretende gravar os exames preparados, bastando para tal, editar o ficheiro em causa.

Também pertencente ao ambiente é o ficheiro que mantém a referência de todos os exames preparados (caminhos no disco rígido), ou seja a *worklist*. O agente *apd* é responsável por editar esta lista, inserindo as novas referências correspondentes aos exames preparados. É de referir ainda que também pertencem ao ambiente do sistema os exames DICOM e os respectivos exames preparados, sem os quais não faria sentido a existência do sistema.

Os acontecimentos associados ao agente de preparação de dados são descritos na Tabela 2.

Tabela 2: Acontecimentos (ou eventos) do agente de preparação de dados

Evento	Descrição
Activa	- Activação do agente.
exame_dicom	- Detecta que um novo exame se encontra disponível no sistema e é iniciado o processo de preparação de dados.
verifica_caminho_exames_interpretados	- A verificação do caminho onde gravar os exames preparados (<i>parsed</i>) é o segundo passo, do processo de preparação de dados.
interpretação_exame	- Depois de conhecer o caminho onde gravar os exames preparados, tem início a interpretação dos dados, o resultado é gravado no directório escolhido.
actualiza_worklist_sistema	- O último passo deste processo consiste na actualização da <i>worklist</i> do sistema.
Desactiva	- Desactivação do agente.

4.2.3.2 O Agente Estação de Controlo, *aec*

O agente estação de controlo é sem dúvida o agente mais complexo do sistema. Este é responsável pela interface entre os utilizadores e o sistema, por verificar os exames da *worklist* e enviar a informação para os terminais de visualização.

Para saber quais os estudos que se encontram presentes na *worklist*, o agente *aec*, utiliza continuamente os seus sensores para avaliar o estado do ambiente, analisando a lista de referências de exames da *worklist*. É também responsável por eliminar exames preparados e ainda remover as suas referências da *worklist*, quando terminada a análise dos exames em causa.

Os utilizadores podem interagir com a interface da estação de controlo para carregar no máximo de duas séries da *worklist*. Que contêm as imagens que serão visualizadas pelo sistema *multidisplay*. A navegação pelas séries, efectuada utilizando *sliders*, e o protocolo de afixação (*layout*) podem ser manipulados pelo utilizador. Todas estas alterações são reflectidas nos terminais de visualização, o que implica, que é necessário enviar mensagens para o ambiente com a informação dos dados da imagem, atributos da imagem (e.g. centro e largura da janela), *layout* e navegação. As mensagens de navegação e *layout* foram agrupadas pois elas são interdependentes, ou seja, o posicionamento das imagens depende da forma como são distribuídas e vice-versa. Foi adaptado o termo, alteração do *layout*, para referir também às alterações provocadas pela navegação.

O agente estação de controlo recebe, continuamente, mensagens originadas nos terminais de visualização com informação sobre o seu estado (e.g. IP, posição e resolução do *display*). Com esta informação é possível enviar correctamente mensagens destinadas aos vários terminais de visualização, mas o agente necessita ainda de verificar, continuamente, o posicionamento dos índices atribuídos a cada terminal de visualização. Os índices permitem aos terminais calcularem o posicionamento correcto das imagens a visualizar, e os seus valores são atribuídos da esquerda para a direita, do topo para baixo. Considere a figura 12, onde são apresentados três terminais e os seus índices de posicionamento correspondentes. Caso novos terminais sejam detectados pelo sistema é necessário recalcular estes índices de acordo com o novo estado do sistema.

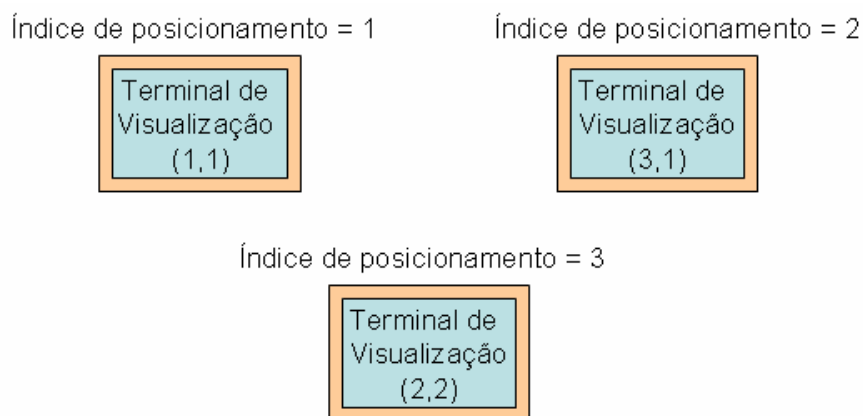


Figura 12: Índices de posicionamento dos terminais de visualização do sistema

Os acontecimentos associados ao agente estação de controlo são descritos na Tabela 3.

Tabela 3: Acontecimentos (ou eventos) do agente estação de controlo

Evento	Descrição
activa	- Activação do agente. Quando o agente é iniciado a lista de terminais de visualização é também iniciada (vazia).
monitoriza_worklist	- Continuamente monitoriza a <i>worklist</i> do sistema. Quando detecta um novo exame actualiza a interface, tornando assim possível a interacção com o mesmo.
termina_analise_exame	- Após terminar a análise do exame a sua referência será retirada da <i>worklist</i> e o exame preparado eliminado do disco rígido.
carregar_série	- Ao efectuar o carregamento de uma série a interface será actualizada e enviados os dados das imagens da série para os terminais de visualização, para que as imagens possam também ser visualizadas nos mesmos. É necessário ainda enviar a informação sobre os atributos da imagem e sobre o <i>layout</i> .
descarregar_série	- Ao descarregar uma série a interface será actualizada e enviada a informação de descarregamento aos terminais para que todos possam eliminar a informação residente em memória RAM e na memória da placa gráfica.
novo_terminal_visualização	- Quando a lista de terminais de visualização for alterada, implica que um novo terminal de visualização foi descoberto pelo sistema. Nesta circunstância é necessário recalcular os novos índices de posicionamento, que afectam a interface da estação de controlo e os terminais de visualização. Após este cálculo é necessário efectuar o recarregamento das séries e enviar a informação sobre o <i>layout</i> e atributos das imagens para os terminais de visualização.
falha_comunicação_rede	- Quando é detectada uma falha de comunicação, a estação de controlo é desactivada e pedido ao utilizador para verificar o estado da sua rede.
alteração_layout	- Quando o <i>layout</i> for alterado, devido à navegação ou efectivamente a uma alteração da disposição das imagens, esta informação irá alterar a interface da estação de controlo e os terminais de visualização. Portanto é necessário enviar esta informação aos terminais de visualização.
alteração_atributos_imagens	- Quando os atributos das imagens de uma série são alterados, estes irão afectar a interface da estação de controlo e os terminais de visualização. Logo é necessário enviar esta informação aos terminais de visualização.
desactiva	- Desactivação do agente. A desactivação deste agente irá provocar a desactivação dos restantes agentes do sistema.

4.2.3.3 Os Agentes Terminais de Visualização, *atv*,

Os agentes terminais de visualização são responsáveis pela apresentação das imagens nos vários *displays* do sistema (excepto estação de controlo).

Como referido anteriormente, estes agentes enviam mensagens para o ambiente sobre o seu estado, i.e., o seu IP, posição e resolução do *display*. E recebem mensagens originadas na estação de controlo com a informação dos dados da imagem, atributos e *layout*. Na mensagem de *layout* encontra-se ainda a informação sobre o índice de posicionamento do terminal de visualização. Utilizando toda esta informação os terminais poderão apresentar as imagens correctamente e de acordo com a estação de controlo.

Os acontecimentos associados aos agentes terminais de visualização são descritos na Tabela 4.

Tabela 4: Acontecimentos (ou eventos) dos agentes terminais de visualização

Evento	Descrição
activa	- Activação do agente.
carregar_série	- Quando a mensagem de carregamento de uma série é detectada pelos terminais, estes irão utilizar esta informação e armazená-la em memória RAM e na memória da placa gráfica, para poderem apresentar correctamente as imagens. Ainda quando é efectuada a carga a interface é alterada para apresentar as imagens da série.
descarregar_série	- Quando a mensagem de descarregamento de uma série é detectada pelos terminais estes irão eliminar a informação residente em memória e a interface é actualizada.
alteração_layout	- Quando a mensagem de alteração do <i>layout</i> é detectada pelos terminais, a interface é actualizada correspondentemente.
alteração_atributos_imagens	- Quando a mensagem de alteração dos atributos das imagens de uma série, é detectada pelos terminais, a interface é actualizada correspondentemente.
informação_terminal_visualização	- A informação do estado dos terminais de visualização é constantemente enviada para o ambiente. A estação de controlo pode utilizar esta informação, para enviar mensagens através do ambiente, para os terminais, correctamente.
desactiva	- Desactivação do agente.

4.2.3.4 O Ambiente *genv*

O ambiente é formado pelos novos estudos/exames da *worklist* do RIS, os exames preparados pelo agente *apd* e um conjunto de ficheiros mantidos nos discos rígidos das máquinas com informação que permite gerir o sistema. Estes ficheiros contêm a seguinte informação:

- Na máquina da estação de controlo:
 - directório dos exames preparados;
 - referências dos exames da *worklist* do sistema (exames preparados).
- Nas máquinas dos terminais de visualização:
 - posicionamento dos *displays* (coluna-linha);
 - IP da estação de controlo;
 - IP do terminal de visualização.

Pertencentes ao sistema são também o quadro negro e todos os canais de comunicação e dispositivos físicos que o compõem.

Tal como descrito em 4.1.2.1 o quadro negro é uma memória partilhada, no nosso caso um processo que corre em *background*, que mantêm e actualiza: os IPs, o posicionamento e a resolução dos terminais de visualização. Utilizando a informação presente no quadro negro, o agente estação de controlo, conhece as características dos terminais e pode comunicar directamente com os agentes terminais de visualização.

4.2.4 Comunicação

A comunicação é realizada através do envio e recepção de mensagens enviadas através do ambiente. Os agentes são responsáveis por sentir estas alterações no ambiente, provocadas pelas mensagens, e responderem de acordo com as mesmas. O formato e descrição das mensagens enviadas são apresentados nas secções seguintes.

4.2.4.1 Mensagens do Agente Estação de Controlo para Quadro Negro

Existem apenas três mensagens enviadas pelo agente da estação de controlo para o quadro negro. O seu formato e descrição são apresentados na tabela 5.

Tabela 5: Mensagens enviadas pelo agente da estação de controlo para o quadro negro

Mensagem	Descrição
<Console>	- Requisita informação ao quadro negro sobre o estado dos terminais de visualização.
<Console reset>	- Limpa (<i>reset</i>) a lista de terminais de visualização existente no quadro negro.
<Console shutdown>	- Desactiva o quadro negro.

4.2.4.2 Mensagens do Quadro Negro para o Agente Estação de Controlo

Após a requisição, por parte da estação de controlo, dos dados existentes no quadro negro, este disponibiliza essa informação de acordo com a mensagem encontrada na tabela 6.

Tabela 6: Mensagens enviadas do quadro negro para o agente estação de controlo

Mensagem	Descrição
<Terminal>< Terminal 1 IP> < Terminal 1 Posição> < Terminal 1 Resolução> :...: <Terminal>< Terminal n IP> < Terminal n Posição> < Terminal n Resolução>	- A mensagem contém a informação dos IPs, posição e resolução de todos os terminais de visualização. A separação é efectuada pelo carácter ":".

4.2.4.3 Mensagens dos Agentes Terminais de Visualização para o Quadro Negro

A informação existente no quadro negro é proveniente dos agentes terminais de visualização. A mensagem correspondente pode ser encontrada na tabela 7.

Tabela 7: Mensagens enviadas dos agentes terminais de visualização para o quadro negro

Mensagem	Descrição
<Terminal><Terminal IP><Terminal Posição><Terminal Resolução>	- A mensagem contém a informação do IP, posição e resolução do terminal de visualização correspondente.

4.2.4.4 Mensagens do Agente Estação de Controlo para os Agentes Terminais de Visualização

As mensagens enviadas do agente estação de controlo para os agentes terminais são apresentadas na tabela 8. Neste caso é mantida uma lista, em memória RAM, contendo as diversas mensagens a enviar. Caso uma mensagem de um determinado tipo não tenha sido ainda processada, e uma nova mensagem do mesmo tipo tenha sido gerada, esta nova mensagem irá para a lista e a mensagem antiga é descartada, evitando assim, o envio desnecessário de mensagens.

Tabela 8: Mensagens enviadas do agente estação de controlo para os agentes terminais de visualização

Mensagem	Descrição
<Layout><Terminal Índice de posicionamento> <Layout Séries> <Layout linhas><Layout colunas> <Série 1 Imagem Inicial> <Série 2 Imagem Inicial>	- A mensagem contém a informação do <i>layout</i> e navegação (posição das imagens iniciais das séries). Contém ainda o índice de posicionamento dos vários terminais.
<Image Attributes > <Série 1 window center> <Série 1 window width> <Série 1 rotation angle > <Série 1 flip vertical > <Série 1 flip horizontal> <Série 1 color inverted> <Série 2 window center> <Série 2 window width> <Série 2 rotation angle > <Série 2 flip vertical > <Série 2 flip horizontal> <Série 2 color inverted>	- A mensagem contém a informação de todos os atributos das imagens das duas séries.
Se Série = 1 <Load Serie S1> Se data type = byte <Série 1 window center> <Série 1 window width> <Série 1 linhas> <Série 1 colunas> <Série 1 data type> valores das densidades globais da série Se data type = short <Série 1 window center> <Série 1 window width> <Série 1 linhas> <Série 1 colunas> <Série 1 data type> valor de controlo da ordem dos bits valores das densidades globais da série Se Série = 2 <Load Serie S2> Se data type = byte <Série 2 window center> <Série 2 window width> <Série 2 linhas> <Série 2 colunas> <Série 2 data type> valores das densidades globais da série Se data type = short <Série 2 window center> <Série 2 window width> <Série 2 linhas> <Série 2 colunas> <Série 2 data type> valor de controlo da ordem dos bits valores das densidades globais da série	- As diversas possibilidades de mensagens de carregamento das séries. No caso das séries, com o tipo de dados <i>short</i> , é necessário enviar um valor de controlo, para recebermos correctamente os valores das densidades, devido à ordem dos bits.
Se Série = 1 <UnLoad Serie S1> Se Série = 2 <UnLoad Serie S2>	- As diversas possibilidades de mensagens de descarregamento das séries.
Se Série = 1 <Load Image S1> <Índice da imagem> valor de controlo da ordem dos bits índices de referência das densidades Se Série = 2 <Load Image L2> <Índice da imagem> valor de controlo da ordem dos bits índices de referência das densidades	- As diversas possibilidades de mensagens de carregamento das imagens das séries. Os valores dos índices de referência das densidades permitem obter a densidade, recorrendo aos valores globais das densidades da série correspondente. Utilizando estes valores é possível calcular as cores de cada <i>pixel</i> da imagem.
<Shutdown>	- Desactiva o terminal de visualização associado.

Capítulo 5

Protótipo do Sistema de Visualização de Imagens Médicas Multidisplay

Com base na especificação do processo de análise de dados médicos e ainda na comparação das funcionalidades e características dos visualizadores existentes, foi desenvolvido um protótipo do sistema de visualização de imagens médicas *multidisplay*. Este protótipo implementa o sistema multiagente apresentado e formalizado no capítulo anterior, garantido a flexibilidade e escalabilidade do sistema global.

Contudo é necessário introduzir os sistemas *multidisplay* para se compreender de que forma podem ser utilizados. Posteriormente será apresentado o protótipo, expondo as suas componentes e funcionalidades.

5.1 Sistemas Multidisplay

Os sistemas *multidisplay* são amplamente utilizados em diversas áreas de aplicação, contudo é na área da Realidade Virtual (RV) que este tipo de sistemas é encontrado com maior frequência. Podem ser utilizados para criar áreas de visualização muito grandes (*large display systems*) como por exemplo *Walls* ou *Domes*. Em [SCHIKORE et al.'00] podemos encontrar um exemplo muito interessante, uma *High-Resolution Multiprojector Display Wall*. Os *displays* encontrados nestes sistemas são, e.g., de dois tipos: baseados em projectores ou monitores. Por exemplo, o protótipo desenvolvido neste trabalho utiliza monitores, devido aos requisitos do sistema, contudo poderiam ser utilizados projectores.

Existem basicamente duas abordagens para a criação de um sistema *multidisplay*. A primeira, a mais simples, recorre ao sistema operativo e ao *hardware* gráfico. Os sistemas operativos actuais possuem funcionalidades que permitem *multidisplay*, bastando configurar o sistema e utilizar placas gráficas com múltiplas saídas e/ou várias placas gráficas. Isto permite que a área de trabalho do sistema operativo cresça de acordo com os monitores utilizados.

A segunda abordagem, mais comum em RV, consiste na utilização de *clusters* de PCs, onde cada PC é conectado a um *display*. Para que estes sistemas funcionem é necessário que exista um meio de comunicação que possibilite o fluxo de informação do sistema global. A baixo nível, e.g., são utilizados *sockets* e os protocolos TCP/IP ou UDP/IP. Na área da RV existem *software frameworks* que gerem estas comunicações a um nível de abstracção superior, exemplos são: o

VR Juggler [WEB-VRJUGGLER'06] e o WireGL [WEB-WIREGL'06]. O VR Juggler para além das comunicações pode ser conectado a *sceengraphs* e a *tracking systems*, desta forma permitindo criar facilmente um sistema de RV complexo. O WireGL foi desenvolvido para permitir a utilização conjunta da API OpenGL pelas várias máquinas que constituem o *cluster*. Actualmente o WireGL possui um sucessor o Chromium [WEB-CHROMIUM'06].

Os sistemas *multidisplay*, baseados em *clusters*, são bastante mais escaláveis e flexíveis, pois não se encontram limitados a um conjunto restrito de *displays* e permitem em alguns casos que exista uma distribuição da carga de processamento (*load balancing*) pelas várias máquinas do *cluster*.

5.2 Estrutura do protótipo

O sistema é composto por dois tipos de elementos distintos, a estação de controlo e os terminais de visualização, como apresentado na figura 13. Estes possuem interfaces gráficas que possibilitam visualizar e interagir com a informação.

No capítulo 4 foram apresentados os agentes pertencentes a estes componentes (*aec* e *atv_i*), que possibilitam que os mesmos possam operar e desempenhar as suas funções correctamente. Agora estes componentes serão apresentados, não ao nível dos seus agentes, mas sim a um nível superior de abstracção onde é dada ênfase às interfaces dos mesmos.

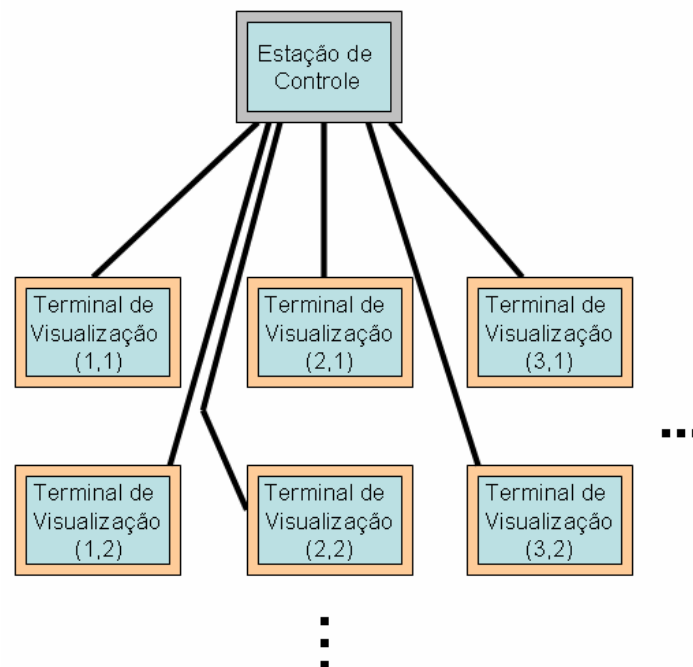


Figura 13: Componentes do Sistema *Multidisplay*

5.2.1 Estação de Controlo

A estação de controlo, ou seja, o agente *aec*, possui ligações a todos os terminais de visualização (agentes *atv*) (figura 13). Ligações dependentes do ambiente do sistema, nomeadamente do quadro negro. A interface da estação de controlo fornece mecanismos de interacção que possibilitam controlar a informação visualizada e disponibilizada aos vários terminais (figura 14). É ainda na estação de controlo que são encontradas ferramentas de análise que auxiliam os utilizadores a analisar detalhadamente a informação. Isto implica a existência de um mecanismo de mudança entre níveis hierárquicos de navegação, que será exaustivamente discutido posteriormente. No entanto podemos constatar, observando a figura 14, que a interface possui um ambiente virtual que contém os terminais de visualização virtuais. No caso apresentado, existem quatro terminais e as imagens presente nos terminais virtuais são idênticas às encontradas nos terminais reais, apenas numa escala diferente. A navegação pelas imagens da série é efectuada utilizando os *sliders* e o protocolo de afixação das séries/imagens é definido utilizando os botões apropriados. Pode ser ainda observado que existem botões para a análise das imagens e uma árvore que contém a *worklist* do sistema. Esta possibilita seleccionar as séries para efectuar o carregamento, utilizando os botões acima da mesma.



Figura 14: Interface da estação de controlo

5.2.2 Terminais de Visualização

Os vários terminais de visualização (*atv*) do sistema possuem uma ligação à estação de controlo (*aec*) (figura 13). Dependendo desta ligação para obterem a informação necessária. Todas as alterações efectuadas na interface da estação de controlo, que influenciem os terminais de visualização, têm de ser imediatamente reportadas. Desta forma garante-se que a informação apresentada nos terminais de visualização virtuais, numa escala reduzida, é idêntica à dos terminais reais, apresentada numa escala muito maior, ocupando toda a área disponível do *display*.

Na figura 15 é apresentada a interface de um terminal de visualização, sendo este terminal correspondente ao terminal de visualização virtual do canto superior esquerdo da interface da estação de controlo (figura 14), comprovando assim que o conteúdo é efectivamente idêntico.

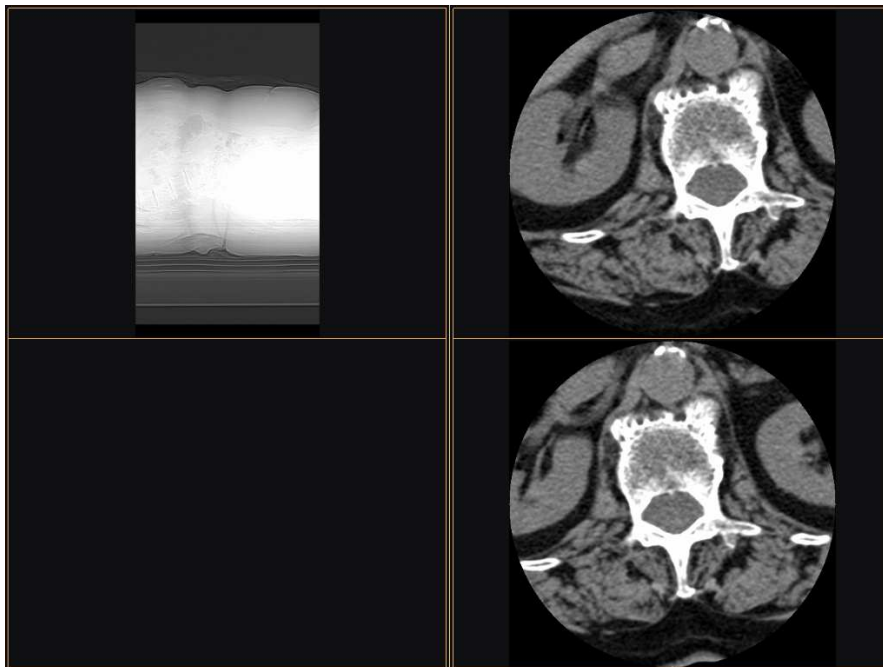


Figura 15: Interface de um terminal de visualização

Para finalizar é apresentada na figura 16 uma fotografia do sistema em funcionamento. Nesta fotografia podemos observar que existem quatro terminais de visualização e uma estação de controlo, onde o conteúdo dos terminais de visualização virtuais é idêntico ao dos terminais reais.



Figura 16: Fotografia do sistema em funcionamento utilizando quatro terminais de visualização

5.3 Protocolo de Afixação

A disposição das imagens pelos vários *displays* é designada por protocolo de afixação. O protocolo de afixação é facilmente controlado utilizando a estação de controlo. O resultado desta interacção pode ser visualizado nos terminais reais e virtuais. Para garantir que a representação dos terminais virtuais seja consistente com os reais é necessário garantir que:

- Os *displays* sejam ordenados como se encontram na realidade;
- As proporções relativas (resolução) dos *displays* sejam mantidas;
- O conteúdo visível na estação de controlo seja idêntico ao apresentado nos *displays* dos terminais de visualização reais.

Quanto ao primeiro ponto, é da responsabilidade do responsável pela instalação do sistema editar o ficheiro `TerminalConfig.txt` existente em cada terminal, que possui a indicação da posição relativa do terminal, sendo o formato o seguinte:

<coluna linha>

De acordo com a secção 4.2.3.4., idealmente os terminais devem ser configurados reflectindo a sua posição real, caso contrário o utilizador poderá ficar desorientado.

O segundo ponto é determinado automaticamente, bastando verificar as propriedades de resolução do *display*, utilizando as funcionalidades do wxWidgets para obter o valor. Estas informações são recolhidas pelos agentes dos terminais de visualização e reportadas à estação de controlo como referido na secção 4.2.4.3.

O último ponto é garantido através de uma disposição correcta das imagens pelos terminais. Foi utilizada uma abordagem de disposição de imagens semelhante à do eFilmTM, onde podemos especificar a disposição das séries pelos *displays* e num nível inferior as imagens dentro da área disponível para as séries. Na figura 17 são apresentados os componentes utilizados para controlar o protocolo de afixação.

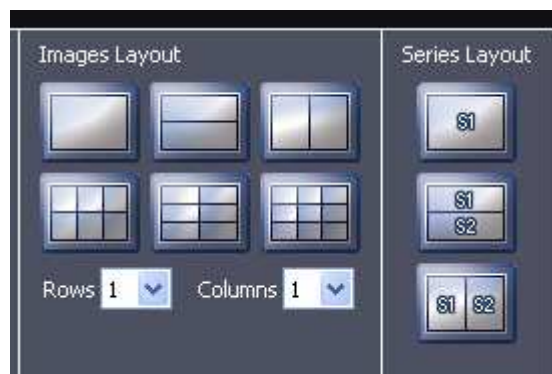


Figura 17: Componentes de controlo do protocolo de afixação

Existem duas diferenças fundamentais a destacar entre a abordagem utilizada neste trabalho e a do eFilmTM. A primeira diferença diz respeito ao número de séries que podem ser visualizadas, ao mesmo tempo, no nosso caso apenas duas (S1 e S2). E a segunda diferença, sem dúvida a mais marcante, deve-se ao facto de existir um nível adicional no nosso sistema, o nível dos terminais de visualização.

Existem três passos necessários para garantir que o protocolo de afixação seja efectuado de acordo com a abordagem descrita anteriormente, são eles:

- Dividir o ambiente virtual numa grelha em que cada célula poderá conter dentro de si um terminal de visualização.
- Dividir as áreas atribuídas aos diversos terminais de visualização também numa grelha onde cada célula pode conter uma série.
- Dividir as áreas atribuídas às diversas séries também numa grelha onde cada célula pode conter uma imagem.

Seguidamente, serão apresentadas as formas de cálculo utilizadas para obter as várias áreas atribuídas.

5.3.1 Cálculo das áreas dos Terminais de Visualização

Inicialmente é necessário dividir o ambiente virtual numa grelha onde cada célula contém dentro de si um terminal de visualização. Na figura 18 podemos observar esta grelha, neste caso particular possui três colunas e duas linhas. Os terminais de visualização possuem dimensões diferentes, isto porque, a sua resolução é diferente. Desta forma é mantida a proporcionalidade relativa entre os terminais. Em relação ao posicionamento, podemos observar que podem existir células que não possuem terminais de visualização atribuídos. O posicionamento é definido pelos ficheiros de configuração dos terminais.



Figura 18: Áreas atribuídas aos *displays*

De seguida serão apresentados os passos necessários para calcular as áreas dos terminais. O primeiro passo consiste em determinar a dimensão das células, através da equação 1.

$$Dc = (Dc_h, Dc_v) = \left(\frac{Dt_h}{Nc}, \frac{Dt_v}{Nl} \right) \quad (1)$$

Onde Nc é o número de colunas, Nl o número de linhas, Dt a dimensão total do ambiente virtual e Dc a dimensão das células.

No passo seguinte, vamos encontrar uma relação que permita que todos os terminais de visualização, possam ser contidos dentro das células e manter proporcionalidade. Para tal, primeiro é necessário verificar quais as maiores resoluções verticais e horizontais $MRes$ e utilizar a equação 2 para obtenção dos rácios verticais e horizontais.

$$MRac = (MRac_h, MRac_v) = \left(\frac{Dc_h}{MRes_h}, \frac{Dc_v}{MRes_v} \right) \quad (2)$$

Para manter a proporcionalidade, é necessário utilizar o menor rácio de $MRac$, de acordo com a equação 3.

$$mMRac = \min(MRac_h, MRac_v) \quad (3)$$

Possuindo este valor, podemos calcular as dimensões das áreas ocupadas pelos terminais segundo a equação 4:

$$Dtv_{(c,l)} = (mMRac \times Res_{h(c,l)}, mMRac \times Res_{v(c,l)}) \quad (4)$$

O último passo, consiste no posicionamento dos terminais de visualização nas suas respectivas células, para que estes se encontram centrados nas mesmas. Para tal calcula-se o centro de cada célula e com esta informação podemos obter as posições dos vértices dos vários terminais. Os centros das células podem ser calculados através da equação 5.

$$Cc_{(c,l)} = \left(\frac{Dc_h}{2} + ((c-1) \times Dc_h), \frac{Dc_v}{2} + ((l-1) \times Dc_v) \right) \quad (5)$$

E consequentemente os vértices que delimitam o rectângulo dos terminais de visualização são obtidos pela equação 6.

$$Vtv_{(c,l)} = \left[\begin{pmatrix} -\frac{Dtv_h}{2}, -\frac{Dtv_v}{2} \\ -\frac{Dtv_h}{2}, \frac{Dtv_v}{2} \end{pmatrix}, \begin{pmatrix} \frac{Dtv_h}{2}, -\frac{Dtv_v}{2} \\ \frac{Dtv_h}{2}, \frac{Dtv_v}{2} \end{pmatrix} \right] + Cc_{(c,l)} \quad (6)$$

Para que as bordas das áreas dos terminais de visualização, não coincidam, com as bordas das células, diminui-se quatro *pixels* em todas as arestas, garantindo uma convergência para o ponto central.

Contudo ainda existe mais uma operação a realizar. Devido ao facto do centro do sistema de coordenadas da área de visualização encontrar-se efectivamente no centro do ambiente virtual, é necessário efectuar uma translação. A equação 7 apresenta esta operação.

$$Vtv' = Vtv - \frac{Dt}{2} \quad (7)$$

5.3.2 Cálculo das áreas das séries

Após o cálculo das áreas dos terminais de visualização, podemos passar ao cálculo das áreas atribuídas às séries. A disposição das séries contempla apenas três casos possíveis, são eles:

- Visualizar apenas uma série;
- Visualizar duas séries dispostas verticalmente;
- Visualizar duas séries dispostas horizontalmente.

No primeiro caso os vértices das áreas são idênticos aos dos terminais virtuais, contudo para que as arestas não se sobreponham são reduzidas de quatro *pixels*, convergindo para o interior. Quanto aos restantes casos, é necessário calcular as duas células das matrizes associadas e reduzir também de quatro *pixels*.

A forma genérica para o cálculo dos vértices das células é dada pela equação 11, sem considerar a redução dos *pixels*.

$$mV_{tv'_{(tv)}} = (mV_{tv'_{(tv)h}}, mV_{tv'_{(tv)v}}) = (\min(V_{tv'_{(tv)h}}), \min(V_{tv'_{(tv)v}})) \quad (8)$$

$$MV_{tv'_{(tv)}} = (MV_{tv'_{(tv)h}}, MV_{tv'_{(tv)v}}) = (\max(V_{tv'_{(tv)h}}), \max(V_{tv'_{(tv)v}})) \quad (9)$$

$$\nabla_{S_{(tv)}} = \left(\frac{MV_{tv'_{(tv)h}} - mV_{tv'_{(tv)h}}}{N_{cs}}, \frac{MV_{tv'_{(tv)v}} - mV_{tv'_{(tv)v}}}{N_{ls}} \right) \quad (10)$$

$$V_{S_{(c,l,tv)}} = \begin{bmatrix} ((c-1) \times \nabla_{S_{(tv)h}}, (l-1) \times \nabla_{S_{(tv)v}}) & (c \times \nabla_{S_{(tv)h}}, (l-1) \times \nabla_{S_{(tv)v}}) \\ ((c-1) \times \nabla_{S_{(tv)h}}, l \times \nabla_{S_{(tv)v}}) & (c \times \nabla_{S_{(tv)h}}, l \times \nabla_{S_{(tv)v}}) \end{bmatrix} + mV_{tv'_{(tv)}} \quad (11)$$

Onde N_{ls} e N_{cs} são respectivamente o número de linhas e colunas das séries.

O resultado gráfico dos três casos possíveis pode ser observado na figura 19. Como as arestas das áreas das imagens (calculadas na secção seguinte) se irão sobrepor às das séries não é necessário desenhá-las, apenas a linha separadora das séries é desenhada.

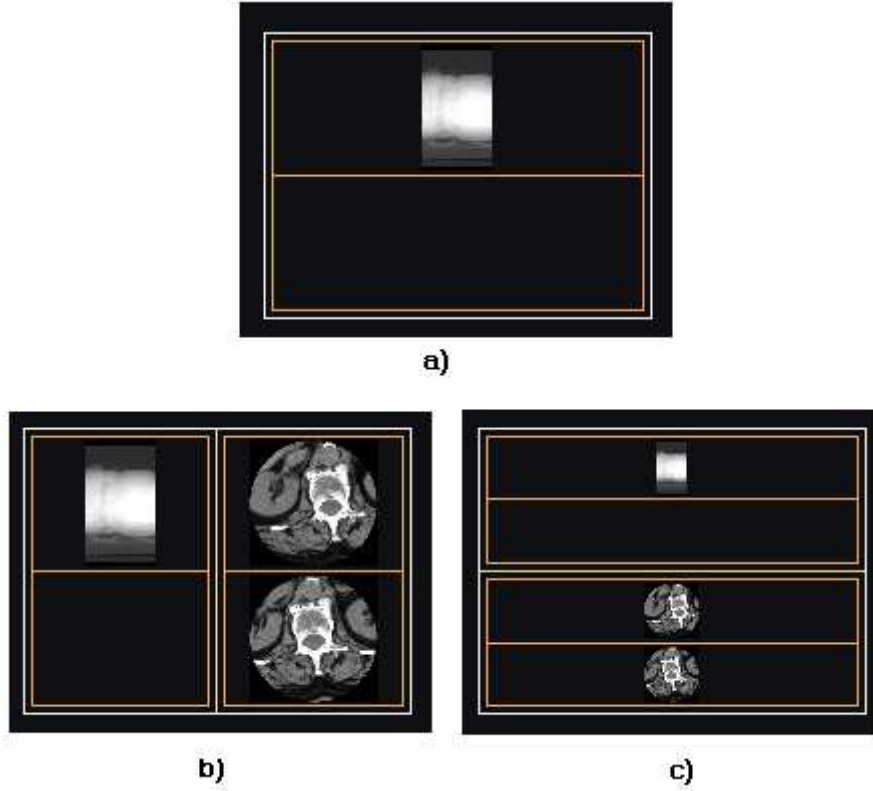


Figura 19: Áreas atribuídas às séries a) apenas série S1, b) duas séries S1 e S2 distribuídas horizontalmente, c) duas séries S1 e S2 distribuídas verticalmente

5.3.3 Cálculo das áreas das imagens

Finalmente falta calcular a distribuição das imagens dentro das áreas das séries. Para tal recorre-se a uma abordagem semelhante à utilizada para o cálculo das áreas das séries. Neste caso, a equação geral que fornece os vértices das células da grelha de imagens é a equação 15.

$$mVs_{(tv,s)} = (mVs_{(tv,s)h}, mVs_{(tv,s)v}) = (\min(Vs_{(tv,s)h}), \min(Vs_{(tv,s)v})) \quad (12)$$

$$MVs_{(tv,s)} = (MVs_{(tv,s)h}, MVs_{(tv,s)v}) = (\max(Vs_{(tv,s)h}), \max(Vs_{(tv,s)v})) \quad (13)$$

$$\nabla i_{(tv,s)} = \left(\frac{MVs_{(tv,s)h} - mVs_{(tv,s)h}}{Nci}, \frac{MVs_{(tv,s)v} - mVs_{(tv,s)v}}{Nli} \right) \quad (14)$$

$$Vi_{(c,l,tv,s)} = \begin{bmatrix} ((c-1) \times \nabla i_{(tv,s)h}, (l-1) \times \nabla i_{(tv,s)v}) & (c \times \nabla i_{(tv,s)h}, (l-1) \times \nabla i_{(tv,s)v}) \\ ((c-1) \times \nabla i_{(tv,s)h}, l \times \nabla i_{(tv,s)v}) & (c \times \nabla i_{(tv,s)h}, l \times \nabla i_{(tv,s)v}) \end{bmatrix} + mVs_{(tv,s)} \quad (15)$$

A figura 20, apresenta um caso onde apenas é visualizada a série S1, e dentro da sua área podemos verificar que existem seis imagens (duas linhas, três colunas). Este é apenas um exemplo das várias configurações possíveis.

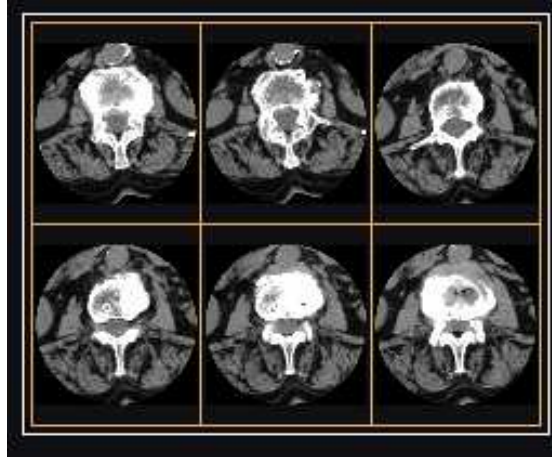


Figura 20: Exemplo de uma distribuição de imagens dentro de uma série

A área atribuída a cada imagem, não é necessariamente, idêntica à área ocupada pela imagem, mas sim uma área que garanta que as imagens possam ser contidas dentro das células. As imagens encontram-se centradas nas suas células e o rácio das imagens é sempre mantido, para que não existam distorções. Isto implica o cálculo das posições e dimensões das imagens de uma forma semelhante ao efectuado no posicionamento das áreas dos terminais de visualização dentro das células da grelha do ambiente virtual. As equações que descrevem este processo são apresentadas seguidamente.

É necessário obter o rácio que garanta que as imagens possam ser posicionadas correctamente. Tal como em 4.3.1 o rácio escolhido é o menor de entre os rácios horizontais e verticais, como apresentado na equação 20.

$$mVi_{(tv,s,i)} = (mVi_{(tv,s,i)h}, mVi_{(tv,s,i)v}) = (\min(Vi_{(tv,s,i)h}), \min(Vi_{(tv,s,i)v})) \quad (16)$$

$$MVi_{(tv,s,i)} = (MVi_{(tv,s,i)h}, MVi_{(tv,s,i)v}) = (\max(Vi_{(tv,s,i)h}), \max(Vi_{(tv,s,i)v})) \quad (17)$$

$$\nabla IDicom_{(tv,s,i)} = (MVi_{(tv,s,i)h} - mVi_{(tv,s,i)h}, MVi_{(tv,s,i)v} - mVi_{(tv,s,i)v}) \quad (18)$$

$$IDicomRac_{(tv,s,i)} = (IDicomRac_{(tv,s,i)h}, IDicomRac_{(tv,s,i)v}) = \left(\frac{\nabla IDicom_{(tv,s,i)h}}{IDicomNc}, \frac{\nabla IDicom_{(tv,s,i)v}}{IDicomNl} \right) \quad (19)$$

$$mIDicomRac = \min(IDicomRac_h, IDicomRac_v) \quad (20)$$

Onde $IDicomNc$ e $IDicomNl$ são respectivamente, o número de colunas e linhas da imagem DICOM.

O passo seguinte consiste no cálculo dos centros e posteriormente dos vértices.

$$CcIDicom_{(tv,s,i)} = \left(mVi_{(tv,s,i)h} + \frac{\nabla IDicom_{(tv,s,i)h}}{2}, mVi_{(tv,s,i)v} + \frac{\nabla IDicom_{(tv,s,i)v}}{2} \right) \quad (21)$$

$$DIDicom_{(tv,s,i)} = \begin{bmatrix} (-IDicomNc, -IDicomNl) & (IDicomNc, -IDicomNl) \\ (-IDicomNc, IDicomNl) & (IDicomNc, IDicomNl) \end{bmatrix} \quad (22)$$

$$VIDicom_{(tv,s,i)} = \frac{mIDicomRac}{2} \times DIDicom_{(tv,s,i)} + CcIDicom_{(tv,s,i)} \quad (23)$$

É ainda necessário reduzir um pouco a dimensão da imagem, para que esta não se sobreponha às linhas das células. Neste caso a abordagem é um pouco diferente da utilizada anteriormente, onde apenas se retira um valor constante nas várias arestas, isto porque, caso se retire um valor constante a imagem pode ficar distorcida. Para não distorcer a imagem, considera-se que a célula onde a imagem se encontra, é ligeiramente mais pequena e assim resolve-se este problema, porque os rácios serão calculados para esta área garantindo assim proporcionalidade.

5.4 Navegação

A interface da estação de controlo possui diversos mecanismos de navegação que permitem intuitivamente aceder e manipular os diversos tipos de informação existentes. Estes mecanismos, permitem ainda que o utilizador altere rapidamente a percepção da informação visualizada, auxiliando assim, o processo de pesquisa e análise. Foram implementados três mecanismos de navegação, discutidos nas secções seguintes.

5.4.1 Navegação pela Worklist

A navegação recorrendo à *worklist* do sistema, permite que o utilizador navegue através da informação relativa aos estudos, como apresentado na figura 21. A *worklist* possui cinco níveis:

- 1 Paciente;
- 2 Modalidade;
- 3 Estudo;
- 4 Série;
- 5 Imagem.

A estrutura adoptada é um pouco diferente da DICOM. Foi introduzido um nível adicional a Modalidade, porque quando são efectuadas pesquisas por estudos/exames este é um dos termos mais relevantes e permite ainda que a informação se encontre numa forma mais legível.

Acima da árvore da *worklist* encontramos os botões que permitem efectuar o carregamento (*load*) e descarregamento (*unload*) das séries escolhidas. Para efectuar o carregamento de uma série, basta, o utilizador seleccionar na árvore a série e depois clicar no botão da série que pretende carregar (S1 ou S2).

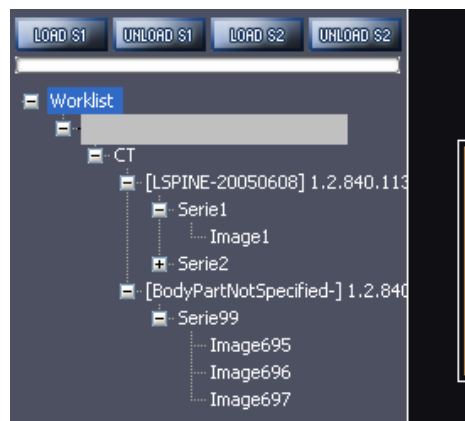


Figura 21: *Worklist* do sistema

5.4.2 Navegação pelas Séries

A navegação através das séries é efectuada utilizando *sliders* (figura 22). Os dois *sliders* da esquerda permitem a navegação através das duas séries. Existe ainda um *slider* adicional para a navegação pela série seleccionada. Na secção seguinte, será explicada a situação onde o último *slider* se torna activo. Existe ainda um botão, o *Sync*, que permite sincronizar as duas séries, desta forma o utilizador ao alterar um *slider* essa alteração é reflectida, correspondentemente, no outro *slider*.

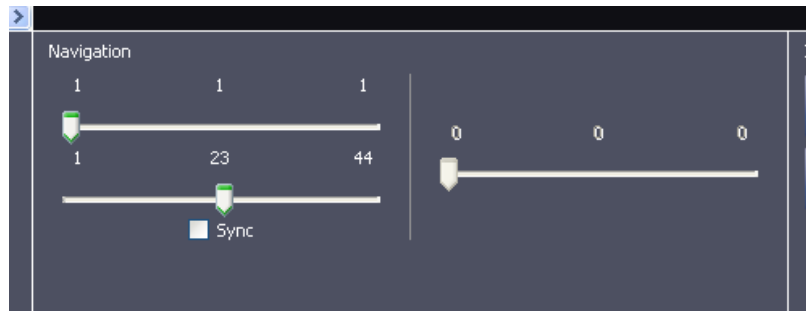


Figura 22: *Sliders* utilizados na navegação pelas séries

5.4.3 Navegação pelos níveis do Ambiente Virtual

A interface da estação de controlo possui um ambiente virtual que contempla diversos níveis de visualização. O nível inferior do ambiente apresenta os terminais de visualização virtuais, proporcionando uma vista geral de todo o conjunto e também dos seus conteúdos, como apresentado na figura 23.



Figura 23: Nível dos terminais de visualização

O conteúdo encontrado nos terminais de visualização reais é idêntico ao encontrado nos terminais de visualização virtuais, contudo existe uma diferença de escala. Nos terminais de visualização reais as imagens são apresentadas numa escala muito superior, porque podem ocupar todo o espaço disponível do *display*. Os terminais virtuais, mantêm a proporcionalidade e o posicionamento, auxiliando assim os utilizadores numa identificação mais clara dos terminais reais correspondentes.

O nível seguinte é o do terminal de visualização seleccionado. Para passar a este nível o utilizador apenas necessita de efectuar um duplo *click* sobre a área correspondente a um terminal de visualização. Após o duplo *click* o utilizador passará a visualizar apenas o terminal seleccionado, como apresentado na figura 24. Para voltar ao nível inicial basta efectuar *Ctrl+click*.

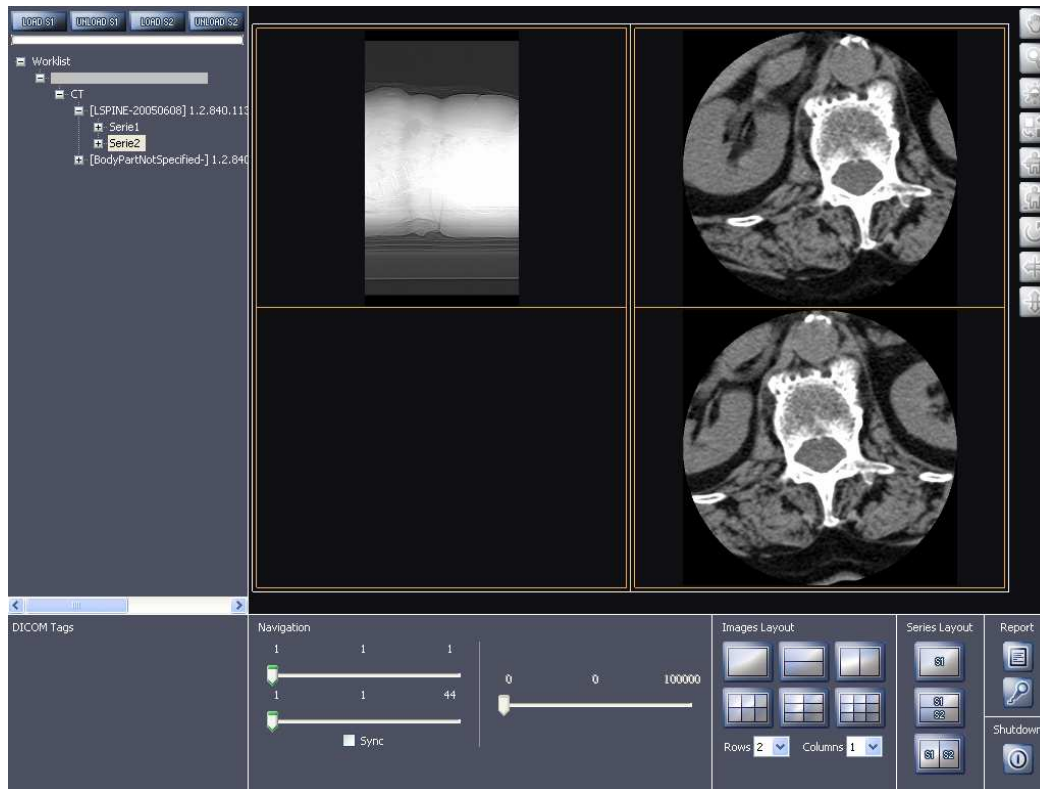


Figura 24: Nível do terminal de visualização seleccionado

O nível superior é o da imagem seleccionada. Para passar a este nível o utilizador efectua novamente um duplo *click* sobre a área correspondente a uma imagem. Após o duplo *click* o utilizador passará a visualizar, apenas, a imagem seleccionada, como apresentado na figura 25. No entanto, existe ainda uma outra forma de passar a este nível, caso o utilizador se encontre no nível dos terminais de visualização e seleccionar um terminal que contenha apenas uma imagem passa directamente para este nível. Para voltar ao nível inferior basta efectuar *Ctrl+click*.

Neste nível o utilizador passa a ter à sua disposição as ferramentas de análise de imagem e o *slider* que se encontra isolado à direita. As ferramentas de análise serão posteriormente apresentadas e discutidas, detalhadamente. O *slider* permite a navegação pelas imagens da série correspondente. A navegação não influencia o estado dos níveis inferiores, é portanto, independente e constitui uma ferramenta muito importante, podendo ser encontrada na maioria dos visualizadores de imagem médica.

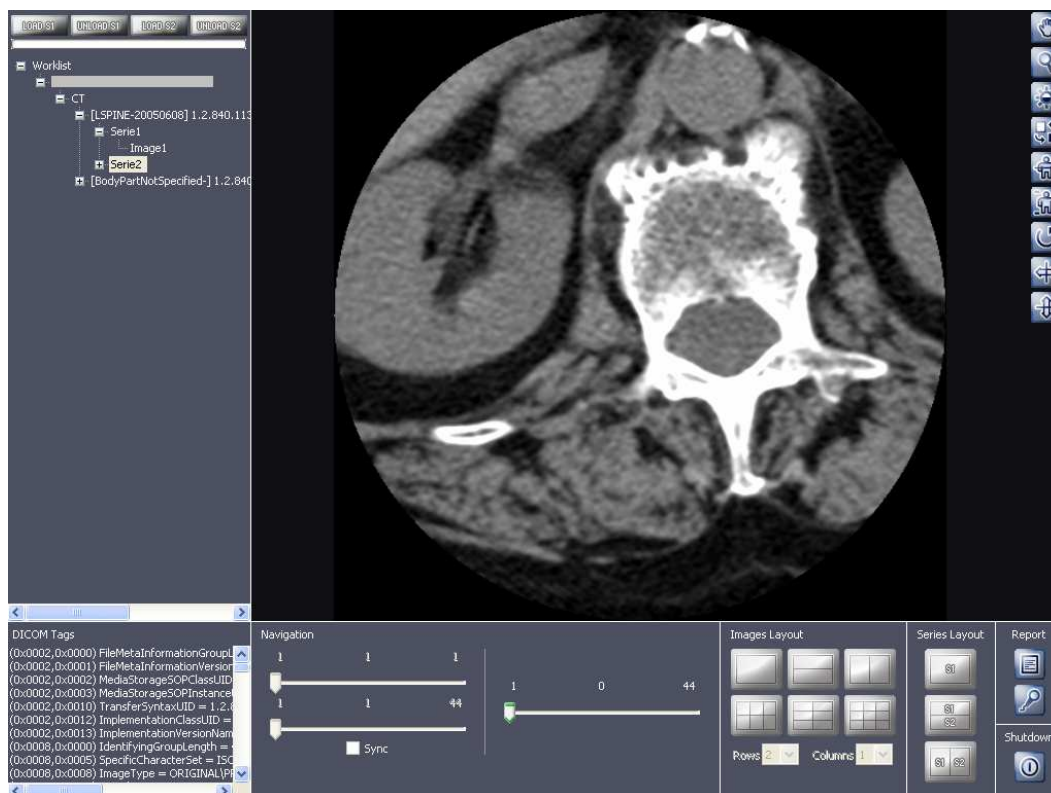


Figura 25: Nível da imagem seleccionada

Para finalizar podemos observar os vários níveis do ambiente virtual na figura 26.

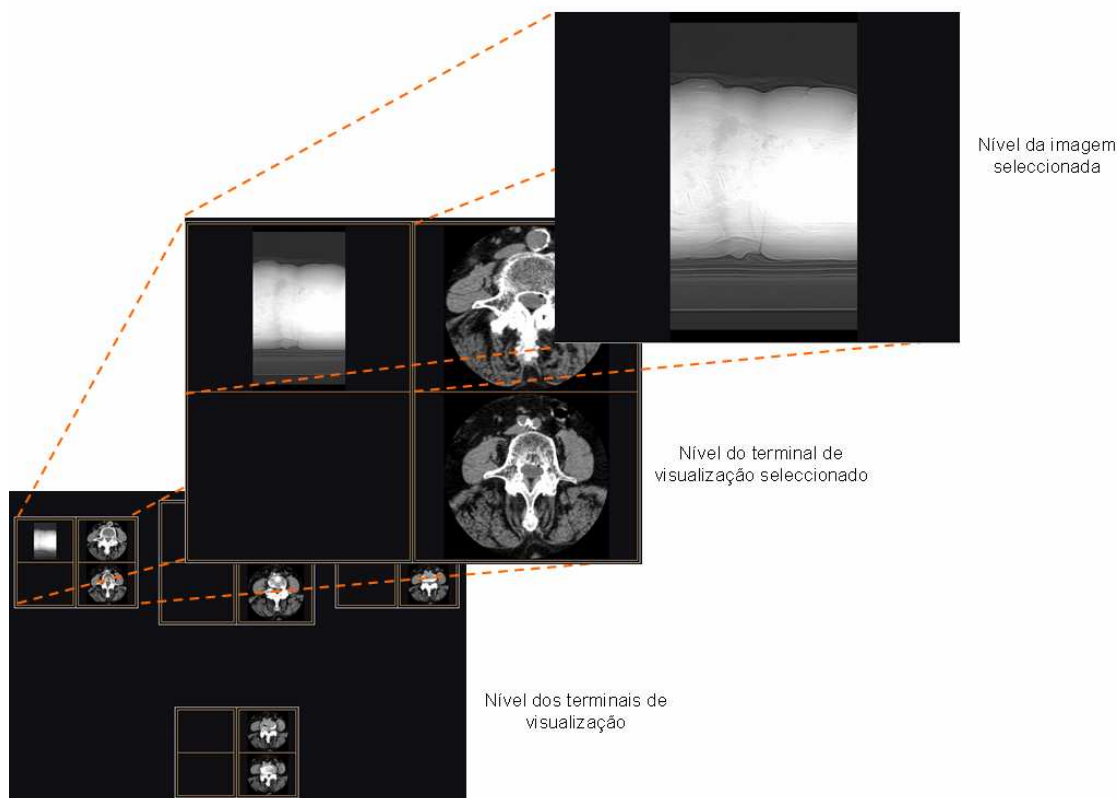


Figura 26: Os vários níveis do ambiente virtual

5.5 Ferramentas de análise

As ferramentas de análise são disponibilizadas quando o utilizador se encontra no nível da imagem seleccionada, como apresentado na figura 25. Estas ferramentas são as mais utilizadas e encontram-se disponíveis na maioria dos visualizadores de imagens médicas. A barra de ferramentas é apresentada na figura 26. De seguida estas ferramentas serão analisadas, detalhadamente.



Figura 27: Barra de ferramentas

5.5.1 Translação (Pan)

A translação, também conhecida por *pan*, permite que o utilizador altere a posição da imagem. Na figura 28 podemos ver uma imagem que sofreu uma translação e sinalizado com um círculo vermelho o botão associado a esta função.

Como podemos verificar a imagem não se encontra centrada no centro do ambiente virtual, devido à translação. Para efectuar a translação o utilizador deve clicar no botão correspondente, o que provocará a mudança do cursor do rato para uma mão aberta, seguidamente pode clicar sobre a imagem, o que provocará uma mudança do cursor do rato para uma mão fechada e arrastando a imagem o utilizador pode colocá-la onde pretende.

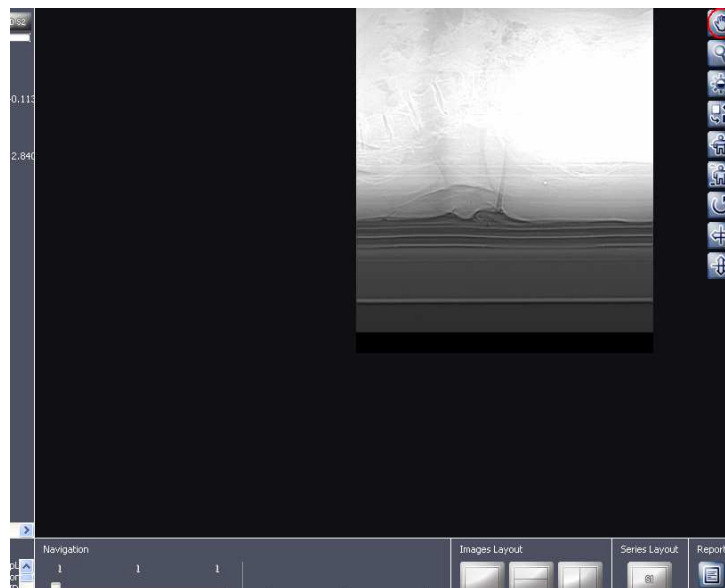


Figura 28: Exemplo - translação

5.5.2 Zoom

O *zoom* permite que o utilizador efectue redimensionamentos da imagem, podendo assim visualizá-la numa dimensão maior ou menor. Na figura 29, podemos ver dois exemplos, onde a mesma imagem se encontra com dimensões diferentes. O botão associado é o segundo a contar do topo. Para efectuar o *zoom* o utilizador clica no botão correspondente o que provoca uma mudança do cursor do rato para uma lupa. Ao arrastar para cima e para baixo a imagem é aumentada e diminuída respectivamente.

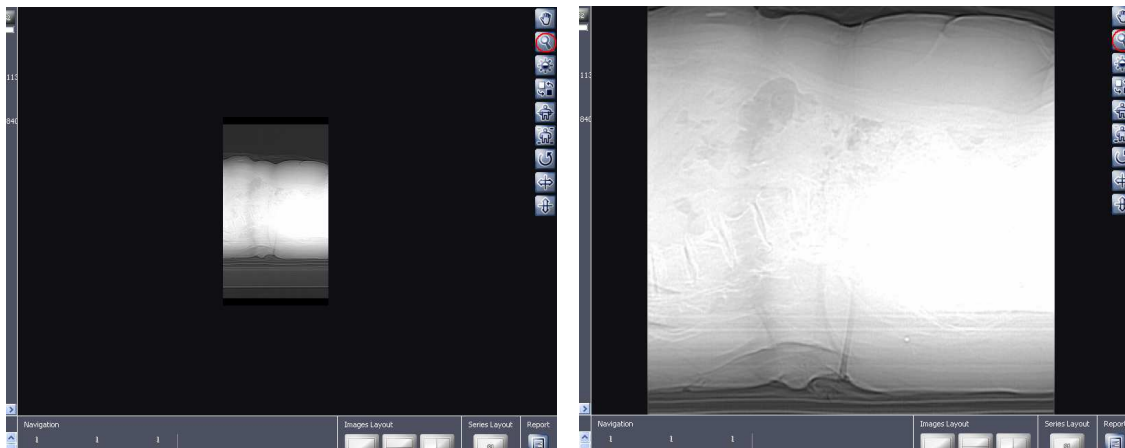


Figura 29: Exemplo - zoom

5.5.3 Controlo do centro e largura da janela

O controlo do centro e largura da janela é uma ferramenta que permite de uma forma expedita controlar o brilho e o contraste das imagens. No caso das imagens TC e RM as cores das imagens dependem dos valores das densidades dos materiais originais. A janela permite seleccionar uma gama restrita destes valores de densidade. O centro da janela corresponde ao

valor da densidade central e a largura representa a diferença entre a maior e a menor densidades.

A figura 30 apresenta dois exemplos da aplicação, à mesma imagem, do controlo do centro e largura da janela. O botão associado é o terceiro a contar do topo. Para efectuar o controlo da janela o utilizador necessita clicar no botão correspondente o que provocará a mudança do cursor do rato para uma imagem idêntica ao do botão. Ao arrastar para cima e para baixo, o utilizador muda o centro da janela. E ao arrastar para a esquerda e direita é alterada a largura da janela. Estas alterações são reflectidas em todas as imagens da série correspondente.

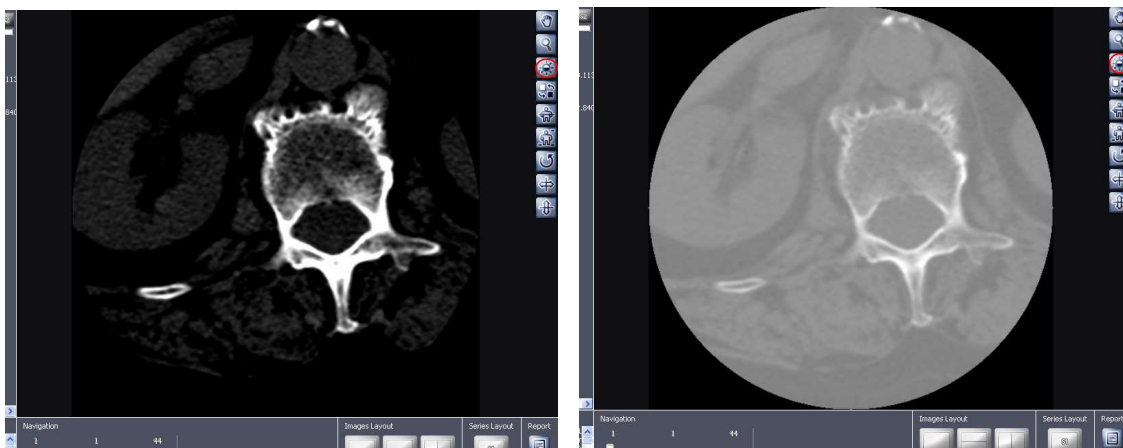


Figura 30: Exemplo - controlo do centro e largura da janela

5.5.4 Inversão de cor

A inversão de cor permite apresentar o negativo da imagem utilizada. Na figura 31 pode ser analisada a imagem original (esquerda) e a sua imagem invertida (direita). O botão associado é o quarto a contar do topo. Para aplicar e reverter a inversão basta clicar sobre o botão correspondente. Esta alteração é reflectida em todas as imagens da série correspondente.

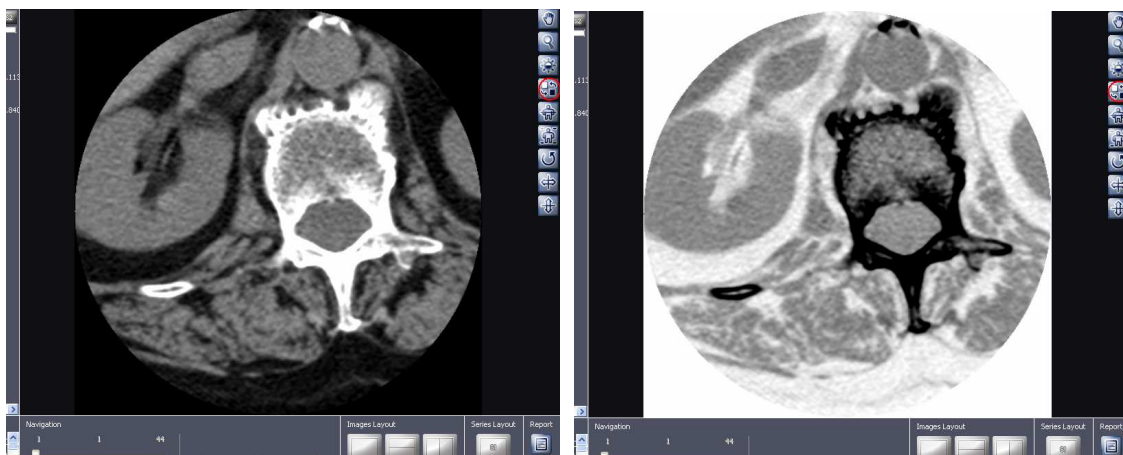


Figura 31: Exemplo – inversão da cor

5.5.5 Imagem de referência

Esta ferramenta é sem dúvida a mais complexa. Ela permite que a imagem seleccionada seja referenciada utilizando uma linha azul sobre uma imagem designada na norma DICOM de *Localizer* (apenas utilizada nas imagens TC). Esta imagem na modalidade TC representa a área global de análise e irá ser designada por imagem de referência neste trabalho. Geralmente esta imagem encontra-se isolada na primeira série do estudo. Na figura 32 podemos observar esta referenciação, onde a posição da imagem seleccionada corresponde à linha azul sobre a imagem de referência (imagem anatômica – ver secção 2.4). O botão associado é o quinto a contar do topo (círculo vermelho). Para apresentar a referenciação basta clicar sobre o botão correspondente.

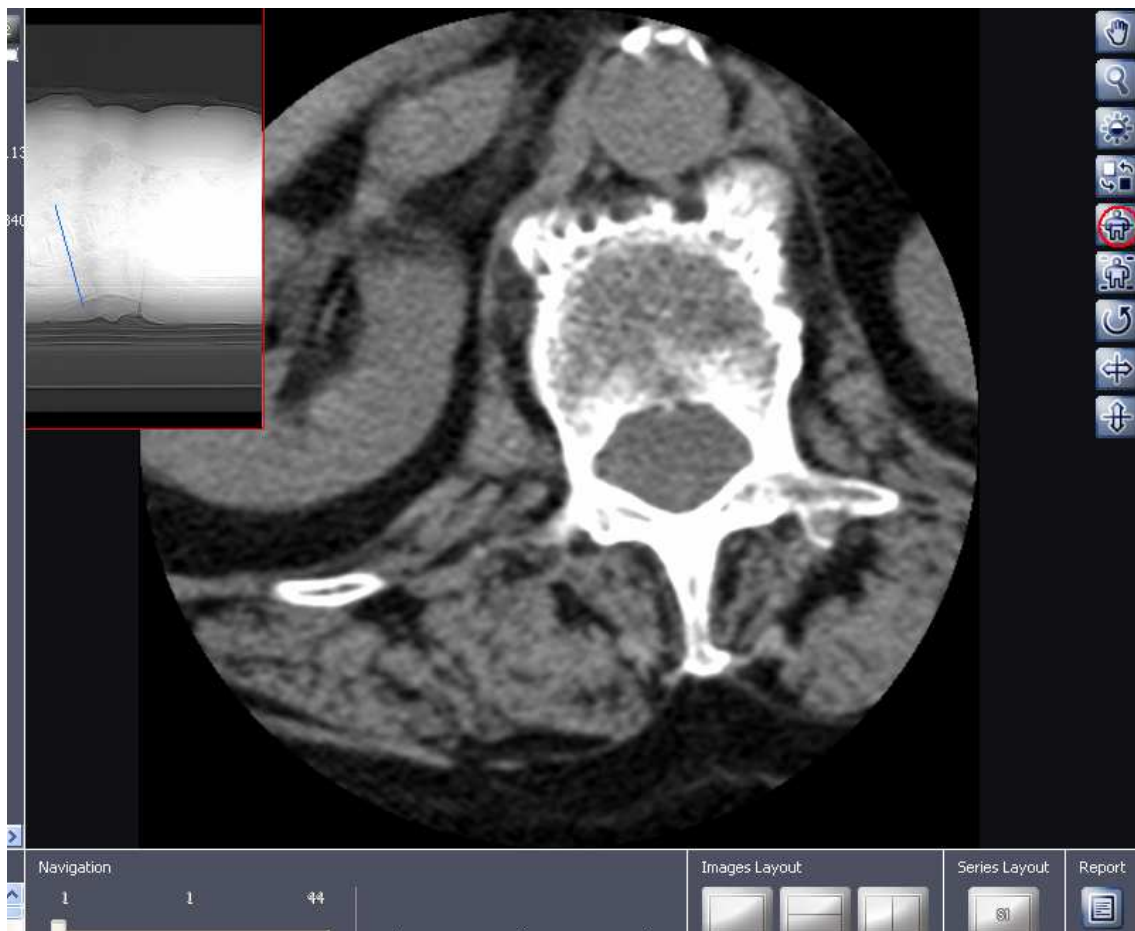


Figura 32: Exemplo – referenciação

O cálculo desta referenciação é matematicamente complexo e será descrito seguidamente.

De acordo com a norma DICOM, parte 3 - secção 7.6.2.1 a matriz que define o posicionamento dos *voxels* das imagens no espaço é apresentada na equação 24.

$$\begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix} = \begin{bmatrix} X_x \Delta i & Y_x \Delta j & 0 & S_x \\ X_y \Delta i & Y_y \Delta j & 0 & S_y \\ X_z \Delta i & Y_z \Delta j & 0 & S_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i \\ j \\ 0 \\ 1 \end{bmatrix} = M \begin{bmatrix} i \\ j \\ 0 \\ 1 \end{bmatrix} \quad (24)$$

onde:

- $P_{(x,y,z)}$ - são as coordenadas do *voxel* (i, j) no plano da imagem em mm;
- $S_{(x,y,z)}$ - são os três valores da *Image Position (Patient)* DICOM Tag (0020,0032). Corresponde à localização inicial da série, i.e. a posição da primeira imagem da série, também em mm;
- $X_{(x,y,z)}$ - são os valores da linha (X) que corresponde a um dos cosenos directores, *Image Orientation (Patient)* DICOM Tag (0020,0037);
- $Y_{(x,y,z)}$ - são os valores da coluna (Y) que corresponde a um dos cosenos directores, *Image Orientation (Patient)* DICOM Tag (0020,0037);
- i - índice das colunas do plano da imagem, o primeiro índice é 0;
- j - índice das linhas do plano da imagem, o primeiro índice é 0;
- Δi - resolução de *pixel* da coluna, *Pixel Spacing* DICOM Tag (0028,0030) em mm;
- Δj - resolução de *pixel* da linha, *Pixel Spacing* DICOM Tag (0028,0030) em mm.

Para conhecermos onde a imagem de referência coincide com a imagem que está a ser analisada, basta verificar os pontos coincidentes recorrendo às matrizes M das duas imagens. Contudo a utilização de um algoritmo que percorra todas as posições dos *voxels* de uma imagem e as comparar com as posições de todos os *voxels* da outra imagem não é eficiente. Poderíamos pensar também em utilizar o cálculo matricial para resolver o sistema, contudo a matriz M não permite inversa logo não existe uma solução imediata. Então para garantir eficiência foi desenvolvido um algoritmo que tira partido das ferramentas matemáticas/geométricas como a intersecção de planos e linhas.

O primeiro passo deste algoritmo consiste na intersecção dos dois planos das imagens. Mas primeiro é necessário obter as normais dos planos que podem ser calculadas utilizando o produto vectorial dos vectores dos cosenos directores ($X_{(x,y,z)}$ e $Y_{(x,y,z)}$) segundo a equação 25.

$$\begin{bmatrix} N_x \\ N_y \\ N_z \end{bmatrix} = X \times Y = \begin{bmatrix} X_y Y_z - X_z Y_y \\ X_z Y_x - X_x Y_z \\ X_x Y_y - X_y Y_x \end{bmatrix} \quad (25)$$

É necessário ainda normalizar a normal segundo a equação 27.

$$\nabla N = \sqrt{N_x^2 + N_y^2 + N_z^2} \quad (26)$$

$$\begin{bmatrix} N_x \\ N_y \\ N_z \end{bmatrix} = \frac{1}{\nabla N} \begin{bmatrix} N_x \\ N_y \\ N_z \end{bmatrix} \quad (27)$$

Conhecendo as normais dos dois planos das imagens (N - Imagem de Referência e N' - Imagem Seleccionada) podemos efectuar a intersecção dos planos [WEB-PLANE-PLANEINTERSECTION'05]. Os dois planos ao intersectarem-se formam uma linha/recta desde que não sejam paralelos. Isto implica que a linha de intersecção tem de ser perpendicular às duas normais N e N' , o que significa que é paralela a pRa .

$$pRa = N \times N' = \begin{bmatrix} N_y N'_z - N_z N'_y \\ N_z N'_x - N_x N'_z \\ N_x N'_y - N_y N'_x \end{bmatrix} \quad (28)$$

Para obtermos a equação da linha é necessário encontrar um ponto (p_0) que pertença simultaneamente aos dois planos, i.e. que satisfaça as equações 29 e 30.

$$N \cdot p_0 = -p_1 \quad (29)$$

$$N' \cdot p_0 = -p_2 \quad (30)$$

Em geral este sistema é indeterminado, mas uma solução particular pode ser obtida, se assumirmos $p_{z0} = 0$. Contudo isto implica que a componente z de pRa não seja nula, o que implica que é necessário considerar todas as possibilidades, i.e. considerar todas as combinações onde N , N' e pRa contêm valores nulos.

Assim vamos expor partes do pseudocódigo para que se entenda o algoritmo utilizado, para calcular o ponto $p_{0(x,y,z)}$:

```
PD' = -(S'x*N'x+S'y*N'y+S'z*N'z);
PD = -(Sx*Nx+Sy*Ny+Sz*Nz);

aNull = verdadeiro;

//#1
se(pRaz diferente de 0)
{
    p0z=0;
    //1
    se((Nx diferente de 0) e (Ny diferente de 0) e
        (N'x diferente de 0) e (N'y diferente de 0))
```

```

{
  se((N'y*Nx-Ny*N'x diferente de 0))
  {
    p0y=(PD*N'x-PD'*Nx)/(N'y*Nx-Ny*N'x);
    p0x=(-Ny*p0y-PD)/Nx;
    aNull = falso;
  }
}

...

//15
c.c. se((Nx diferente de 0) e (Ny igual a 0) e
(N'x igual a 0) e (N'y igual a 0))
{
  p0x=-PD/Ny;
  p0y=0;
  aNull = falso;
}
//16
c.c. se((Nx igual a 0) e (Ny igual a 0) e
(N'x igual a 0) e (N'y igual a 0))
{
  //nada acontece porque as normais não podem ser (0,0,0)
}
}
//#2
se((pRay diferente de 0) e (aNull é verdadeiro))
{
  p0y=0;
  //1
  se((Nx diferente de 0) e (Nz diferente de 0) e
(N'x diferente de 0) e (N'z diferente de 0))
  {
    se((N'z*Nx-Nz*N'x diferente de 0))
    {
      p0z=(PD*N'x-PD'*Nx)/(N'z*Nx-Nz*N'x);
      p0x=(-Nz*p0z-PD)/Nx;
      aNull = falso;
    }
  }
}

...

//15
c.c. se((Nx diferente de 0) e (Nz igual a 0) e
(N'x igual a 0) e (N'z igual a 0))
{
  p0x=-PD/Nx;
  p0z=0;
  aNull = falso;
}
//16
c.c. se((Nx igual a 0) e (Nz igual a 0) e
(N'x igual a 0) e (N'z igual a 0))
{
  //nada acontece porque as normais não podem ser (0,0,0)
}
}
//#3
se((pRax diferente de 0) e (aNull é verdadeiro))
{

```

```

p0x=0;
//1
se((Ny diferente de 0) e (Nz diferente de 0) e
  (N'y diferente de 0) e (N'z diferente de 0))
{
  se((N'z*Ny-Nz*N'y diferente de 0))
  {
    p0z=(PD*N'y-PD'*Ny)/(N'z*Ny-Nz*N'y);
    p0y=(-Nz*p0z-PD)/Ny;
    aNull = falso;
  }
}
...
//15
c.c. se((Ny diferente de 0) e (Nz igual a 0) e
  (N'y igual a 0) e (N'z igual a 0))
{
  p0y=-PD/Ny;
  p0z=0;
  aNull = falso;
}
//16
c.c. se((Ny igual a 0) e (Nz igual a 0) e
  (N'y igual a 0) e (N'z igual a 0))
{
  //nada acontece porque as normais não podem ser (0,0,0)
}
}

se (aNull é verdadeiro)
{
  Os planos são paralelos não pode ser calculada a intersecção.
}

```

Como se pode constatar existem muitas possibilidades a considerar, mas no final obtemos o ponto $p_{0(x,y,z)}$, i.e. caso os planos se interceptem. E assim ficamos a conhecer a equação da linha de referência, que é dada pela equação 31.

$$pR = p_0 + t \cdot pRa \quad (31)$$

Contudo a linha encontra-se delimitada pelos limites (arestas) das imagens. Isto implica que é necessário encontrar os valores de t que permitam definir o segmento de recta interior. Este processo envolve alguns passos que serão apresentados de seguida.

O primeiro passo consiste na determinação dos pontos de intersecção da linha de referência com as linhas delimitadoras da imagem seleccionada, como apresentado na figura 33.

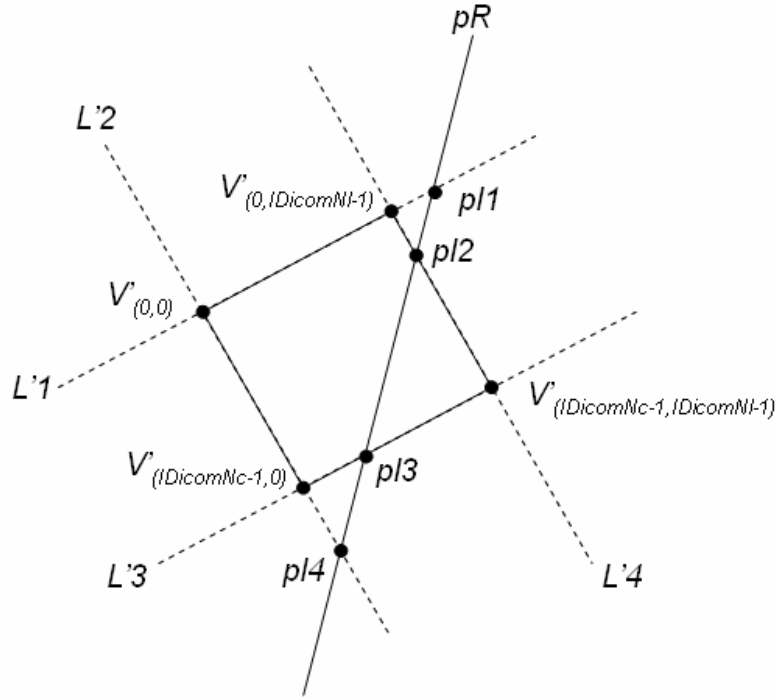


Figura 33: Exemplo da intersecção de uma linha de referência com as linhas limite do plano da imagem

Para definirmos as linhas delimitadoras podemos recorrer aos vértices da imagem, segundo as seguintes equações.

$$V'_{(0,0)} = \begin{bmatrix} V'_x \\ V'_y \\ V'_z \\ 1 \end{bmatrix} = \begin{bmatrix} X'_x \Delta'i & Y'_x \Delta'j & 0 & S'_x \\ X'_y \Delta'i & Y'_y \Delta'j & 0 & S'_y \\ X'_z \Delta'i & Y'_z \Delta'j & 0 & S'_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (32)$$

$$V'_{(IDicomNc-1,0)} = \begin{bmatrix} V'_x \\ V'_y \\ V'_z \\ 1 \end{bmatrix} = \begin{bmatrix} X'_x \Delta'i & Y'_x \Delta'j & 0 & S'_x \\ X'_y \Delta'i & Y'_y \Delta'j & 0 & S'_y \\ X'_z \Delta'i & Y'_z \Delta'j & 0 & S'_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} IDicomNc-1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (33)$$

$$V'_{(0,IDicomNl-1)} = \begin{bmatrix} V'_x \\ V'_y \\ V'_z \\ 1 \end{bmatrix} = \begin{bmatrix} X'_x \Delta'i & Y'_x \Delta'j & 0 & S'_x \\ X'_y \Delta'i & Y'_y \Delta'j & 0 & S'_y \\ X'_z \Delta'i & Y'_z \Delta'j & 0 & S'_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ IDicomNl-1 \\ 0 \\ 1 \end{bmatrix} \quad (34)$$

$$V'_{(IDicomNc-1, IDicomNl-1)} = \begin{bmatrix} V'_x \\ V'_y \\ V'_z \\ 1 \end{bmatrix} = \begin{bmatrix} X'_x \Delta' i & Y'_x \Delta' j & 0 & S'_x \\ X'_y \Delta' i & Y'_y \Delta' j & 0 & S'_y \\ X'_z \Delta' i & Y'_z \Delta' j & 0 & S'_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} IDicomNc-1 \\ IDicomNl-1 \\ 0 \\ 1 \end{bmatrix} \quad (35)$$

Onde $IDicomNc$ e $IDicomNl$ são respectivamente o número de colunas e linhas da imagem DICOM.

$$L'1 = V'_{(0,0)} + (V'_{(IDicomNc-1,0)} - V'_{(0,0)})s \quad (36)$$

$$L'2 = V'_{(0,0)} + (V'_{(0,IDicomNl-1)} - V'_{(0,0)})s \quad (37)$$

$$L'3 = V'_{(0,IDicomNl-1)} + (V'_{(IDicomNc-1,IDicomNl-1)} - V'_{(0,IDicomNl-1)})s \quad (38)$$

$$L'4 = V'_{(IDicomNc-1,0)} + (V'_{(IDicomNc-1,IDicomNl-1)} - V'_{(IDicomNc-1,0)})s \quad (39)$$

Após a definição das linhas delimitadoras podemos calcular os pontos de intersecção, recorrendo às equações de intersecção de linhas [WEB-LINE-LINEINTERSECTION'05]. Este cálculo implica que os pontos pertençam simultaneamente às linhas $L'*$ e pR , ou seja que resolvam um sistema como o apresentado na equação 40.

$$\begin{cases} x = x_1 + (x_2 - x_1)s \\ x = x_3 + (x_4 - x_3)t \end{cases} \quad (40)$$

Em conjunto com a condição de que os quatro pontos (no nosso caso correspondem aos vértices da imagem V'^*) sejam coplanares, equação 41.

$$\begin{vmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \\ x_4 & y_4 & z_4 & 1 \end{vmatrix} = (x_3 - x_1) \cdot [(x_2 - x_1) \times (x_4 - x_3)] \quad (41)$$

Assim podemos obter o valor de s através da equação 42.

$$s = \frac{(c \times b) \cdot (a \times b)}{|a \times b|^2} \quad (42)$$

Onde:

$$a = x_2 - x_1 \quad (43)$$

$$b = x_4 - x_3 \quad (44)$$

$$c = x_3 - x_1 \quad (45)$$

Sabendo s , o ponto de intersecção x é calculado directamente utilizando a equação 46.

$$x = x_1 + a \frac{(c \times b) \cdot (a \times b)}{|a \times b|^2} = x_1 + (x_2 - x_1)s \quad (46)$$

Recorrendo a este formalismo matemático é possível determinar facilmente os pontos de intersecção ($pI1, pI2, pI3, pI4$).

O passo seguinte consiste em verificar quais os pontos que se encontram dentro da área da imagem (no nosso caso sobre os limites da imagem). Considerando a figura 33 podemos observar que os pontos $pI1$ e $pI4$ não se encontram dentro da área da imagem. Para efectuar esta validação é necessário recorrer a vectores auxiliares. Comparando as magnitudes e direcções destes vectores pode-se inferir sobre o posicionamento dos pontos. Vamos apenas exemplificar o processo utilizando o ponto $pI1$, porque é idêntico nos restantes casos.

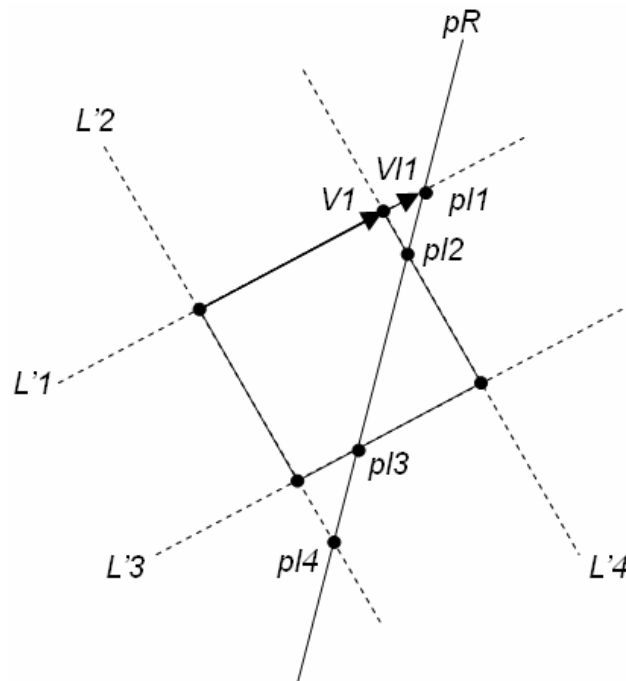


Figura 34: Representação dos dois vectores auxiliares $V1$ e $VI1$

Considerando a figura 34, podemos observar os dois vectores auxiliares $V1$ e $VI1$, definidos segundo as equações seguintes.

$$V1 = V'_{(IDicomNc-1,0)} - V'_{(0,0)} \quad (47)$$

$$VI1 = pI1 - V'_{(0,0)} \quad (48)$$

Para que o ponto seja interior é necessário satisfazer simultaneamente duas condições. A primeira consiste na comparação das magnitudes dos vectores, caso $VI1$ possua uma magnitude superior a $V1$ o ponto encontra-se no exterior, caso contrário pode ser interior. Como se pode observa o vector $VI1$ possui uma magnitude superior a $V1$ logo o ponto $pI1$ é exterior. A segunda condição consiste na comparação da direcção dos vectores, caso os vectores não se encontrem na mesma direcção o ponto é exterior, caso contrário pode ser interior. Para testar esta última condição podemos recorrer aos cosenos directores. Genericamente as condições descritas anteriormente podem ser descritas matematicamente como apresentado nas equações 53 e 54.

$$|V^*| = \sqrt{V_x^{*2} + V_y^{*2} + V_z^{*2}} \quad (49)$$

$$|VI^*| = \sqrt{VI_x^{*2} + VI_y^{*2} + VI_z^{*2}} \quad (50)$$

$$NV^* = \left[\cos \alpha = \frac{V_x^*}{|V^*|}, \quad \cos \beta = \frac{V_y^*}{|V^*|}, \quad \cos \phi = \frac{V_z^*}{|V^*|} \right] \quad (51)$$

$$NVI^* = \left[\cos \alpha = \frac{VI_x^*}{|VI^*|}, \quad \cos \beta = \frac{VI_y^*}{|VI^*|}, \quad \cos \phi = \frac{VI_z^*}{|VI^*|} \right] \quad (52)$$

$$(|VI^*| \leq |V^*|) \wedge (|NVI^*| = |NV^*|) \Rightarrow interior \quad (53)$$

$$(|VI^*| > |V^*|) \vee (|NVI^*| \neq |NV^*|) \Rightarrow exterior \quad (54)$$

Devido aos arredondamentos computacionais estas condições devem ser relaxadas um pouco, ou seja é necessário dar uma certa tolerância às igualdades.

Após estes testes vamos finalmente poder calcular correctamente os valores de t da linha pR que permitem definir o segmento de recta interior. Para tal basta apenas calcular valores de t , dos pontos interiores, utilizando a equação 55, e verificar qual o máximo e mínimo valor de t . Isto porque em alguns casos podemos obter mais do que dois pontos interiores, por exemplo quando a intersecção ocorre num vértice da imagem. Verificando os valores máximo e mínimo de t (t_{min} , t_{max}) assegura-se que a delimitação é correcta.

$$t = \frac{pR_x - p_{0x}}{pRa_x}, pRa_x \neq 0; \quad \vee \quad t = \frac{pR_y - p_{0y}}{pRa_y}, pRa_y \neq 0; \quad \vee \quad t = \frac{pR_z - p_{0z}}{pRa_z}, pRa_z \neq 0; \quad (55)$$

Conhecendo os valores de t que definem a linha interior é possível obter o posicionamento dos dois pontos correspondentes no espaço da imagem de referência. Consideremos o exemplo apresentado da figura 35, onde podemos observar que os dois pontos que delimitaram a linha de referência podem ser facilmente transformados para o espaço bidimensional (i , j) da imagem de referência. Para tal é necessário recorrer às equações que permitem calcular as distâncias entre pontos e linhas [WEB-POINT-LINEDISTANCE2-DIMENSIONAL'05].

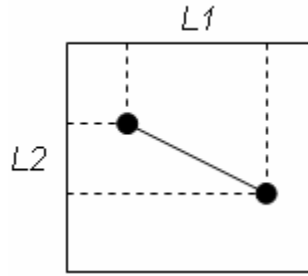


Figura 35: Exemplo da uma linha de referência inserida na imagem de referência

Considerando que os dois pontos são definidos pelas equações 56 e 57.

$$pR_{min} = p_0 + t_{min} \cdot pRa \quad (56)$$

$$pR_{max} = p_0 + t_{max} \cdot pRa \quad (57)$$

E as linhas delimitadoras da imagem $L1$ e $L2$ pelas equações 62 e 63.

$$V_{(0,0)} = \begin{bmatrix} V_x \\ V_y \\ V_z \\ 1 \end{bmatrix} = \begin{bmatrix} X_x \Delta i & Y_x \Delta j & 0 & S_x \\ X_y \Delta i & Y_y \Delta j & 0 & S_y \\ X_z \Delta i & Y_z \Delta j & 0 & S_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (58)$$

$$V_{(IDicomNc-1,0)} = \begin{bmatrix} V_x \\ V_y \\ V_z \\ 1 \end{bmatrix} = \begin{bmatrix} X_x \Delta i & Y_x \Delta j & 0 & S_x \\ X_y \Delta i & Y_y \Delta j & 0 & S_y \\ X_z \Delta i & Y_z \Delta j & 0 & S_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} IDicomNc-1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (59)$$

$$V_{(0,IDicomNl-1)} = \begin{bmatrix} V_x \\ V_y \\ V_z \\ 1 \end{bmatrix} = \begin{bmatrix} X_x \Delta i & Y_x \Delta j & 0 & S_x \\ X_y \Delta i & Y_y \Delta j & 0 & S_y \\ X_z \Delta i & Y_z \Delta j & 0 & S_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ IDicomNl-1 \\ 0 \\ 1 \end{bmatrix} \quad (60)$$

$$V_{(IDicomNc-1,IDicomNl-1)} = \begin{bmatrix} V_x \\ V_y \\ V_z \\ 1 \end{bmatrix} = \begin{bmatrix} X_x \Delta i & Y_x \Delta j & 0 & S_x \\ X_y \Delta i & Y_y \Delta j & 0 & S_y \\ X_z \Delta i & Y_z \Delta j & 0 & S_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} IDicomNc-1 \\ IDicomNl-1 \\ 0 \\ 1 \end{bmatrix} \quad (61)$$

Onde $IDicomNc$ e $IDicomNl$ são respectivamente o número de colunas e linhas da imagem DICOM.

$$L1 = V_{(0,0)} + (V_{(IDicomNc-1,0)} - V_{(0,0)})s \quad (62)$$

$$L2 = V_{(0,0)} + (V_{(0,IDicomNl-1)} - V_{(0,0)})s \quad (63)$$

O passo seguinte consiste na obtenção das distâncias de pR_{min} e pR_{max} a $L1$ e $L2$, para tal recorre-se ao formalismo matemático do cálculo da distância de um ponto a uma linha, como apresentado na figura 36.

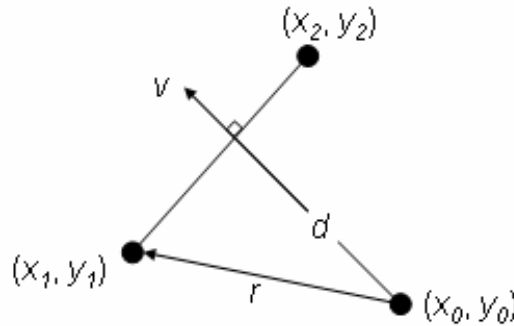


Figura 36: Distância de um ponto a uma linha

O valor de d é obtido segundo a equação 64.

$$d = \frac{|(x_2 - x_1) \times (x_1 - x_0)|}{|(x_2 - x_1)|} \quad (64)$$

Devido ao facto do valor resultante da equação 64 ser sempre um valor positivo não é possível saber se os pontos encontram-se dentro da área da imagem ou fora, tal como apresentado na figura 37.

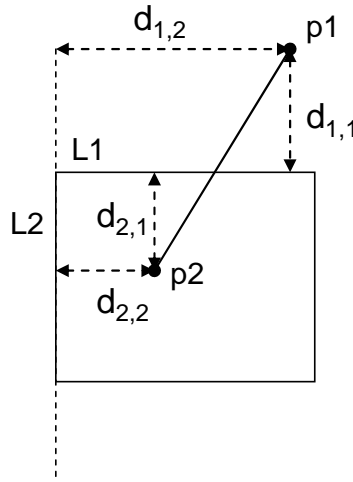


Figura 37: Linha de referência que possui um ponto exterior e um ponto interior

Isto significa que é necessário encontrar o sinal que permita inferir o posicionamento correcto. Por exemplo a distância $d_{1,1}$ (figura 37) deveria ser afectada por um sinal negativo e as restantes distâncias por um sinal positivo.

Para tal é necessário verificar quais os ângulos formados pelas seguintes linhas:

- $L1$ e $Lp = (V(0,0) - p)$;
- $L2$ e $Lp = (V(0,0) - p)$;

Os ângulos podem ser obtidos através do produto escalar dos vectores das rectas ou seja:

$$\alpha_{L1} = \arccos\left(\frac{L1 \cdot Lp}{|L1||Lp|}\right) \quad (65)$$

$$\alpha_{L2} = \arccos\left(\frac{L2 \cdot Lp}{|L2||Lp|}\right) \quad (66)$$

Os sinais respectivos a $L1$ e $L2$ são dados pelas seguintes condições:

$L1$:

$$\begin{aligned} \text{se } (\alpha_{L1} \leq \pi/2) \quad s_j &= 1 \\ \text{c.c.} \quad s_j &= -1 \end{aligned}$$

L2:

$$\begin{aligned} \text{se } (\alpha L2 \leq \pi/2) \quad si &= 1 \\ \text{c.c.} \quad si &= -1 \end{aligned}$$

Utilizando estas equações são calculadas as distâncias de pR_{min} e pR_{max} a $L1$ e $L2$ e obtém-se dois pontos, que serão usados para obter os pontos correspondentes no espaço da imagem. Mas vamos apresentar este cálculo de uma forma explícita utilizando as equações 67 e 68.

$$p_{ref1} = \begin{bmatrix} s_{i \min} \cdot \frac{|(V_{(0, IDicomNl-1)} - V_{(0,0)}) \times (V_{(0,0)} - pR_{\min})|}{|(V_{(0, IDicomNl-1)} - V_{(0,0)})|} \\ s_{j \min} \cdot \frac{|(V_{(IDicomNc-1,0)} - V_{(0,0)}) \times (V_{(0,0)} - pR_{\min})|}{|(V_{(IDicomNc-1,0)} - V_{(0,0)})|} \end{bmatrix} = \begin{bmatrix} i_{ref1} \\ j_{ref1} \end{bmatrix} \quad (67)$$

$$p_{ref2} = \begin{bmatrix} s_{i \max} \cdot \frac{|(V_{(0, IDicomNl-1)} - V_{(0,0)}) \times (V_{(0,0)} - pR_{\max})|}{|(V_{(0, IDicomNl-1)} - V_{(0,0)})|} \\ s_{j \max} \cdot \frac{|(V_{(IDicomNc-1,0)} - V_{(0,0)}) \times (V_{(0,0)} - pR_{\max})|}{|(V_{(IDicomNc-1,0)} - V_{(0,0)})|} \end{bmatrix} = \begin{bmatrix} i_{ref2} \\ j_{ref2} \end{bmatrix} \quad (68)$$

Para obtermos os pontos no espaço da imagem é necessário apenas efectuar uma divisão pelos espaçamentos dos *pixels*. Esta operação é apresentada nas equações 69 e 70.

$$p'_{ref1} = \begin{bmatrix} \frac{i_{ref1}}{\Delta i} \\ \frac{j_{ref1}}{\Delta j} \end{bmatrix} \quad (69)$$

$$p'_{ref2} = \begin{bmatrix} \frac{i_{ref2}}{\Delta i} \\ \frac{j_{ref2}}{\Delta j} \end{bmatrix} \quad (70)$$

Para finalizar esta longa série de operações é necessário ainda verificar se estes pontos encontram-se dentro dos limites da imagem e se existe a necessidade de efectuar *clipping*. Para efectuar o teste podemos recorrer aos valores da dimensão da imagem. Caso os dois pontos se encontrem no interior não é necessário efectuar mais operações, os pontos são válidos e serão utilizados. Caso um ou ambos os pontos se encontrem no exterior é necessário verificar se estes atravessam a área da imagem, caso afirmativo é necessário efectuar o *clipping*. Seguindo o princípio do algoritmo de *clipping* de linhas Cohen-Sutherland vamos dividir o nosso espaço em 9 regiões, como apresentado na figura 38.

1001	1000	1010
0001	0000	0010
0101	0100	0110

Figura 38: Divisão do espaço e respectivos valores binários atribuídos

Os valores binários associados às regiões indicam o seu posicionamento, correspondendo a:

- *bit 1*: esquerda;
- *bit 2*: direita;
- *bit 3*: baixo;
- *bit 4*: cima.

Os pontos de intersecção da linha com as bordas correspondentes às regiões, são definidos pelas equações 71-74.

$$pIE = \begin{bmatrix} pIE_x \\ pIE_y \end{bmatrix} = \begin{bmatrix} 0 \\ p1_y - p1_x \frac{(p2_y - p1_y)}{(p2_x - p1_x)} \end{bmatrix} \quad (71)$$

$$pID = \begin{bmatrix} pID_x \\ pID_y \end{bmatrix} = \begin{bmatrix} IDicomNc - 1 \\ p1_y + (IDicomNc - 1 - p1_x) \frac{(p2_y - p1_y)}{(p2_x - p1_x)} \end{bmatrix} \quad (72)$$

$$pIC = \begin{bmatrix} pIC_x \\ pIC_y \end{bmatrix} = \begin{bmatrix} p1_x - p1_y \frac{(p2_x - p1_x)}{(p2_y - p1_y)} \\ 0 \end{bmatrix} \quad (73)$$

$$pIB = \begin{bmatrix} pIB_x \\ pIB_y \end{bmatrix} = \begin{bmatrix} p1_x + (IDicomNI - 1 - p1_y) \frac{(p2_x - p1_x)}{(p2_y - p1_y)} \\ IDicomNI - 1 \end{bmatrix} \quad (74)$$

Onde $p1$ e $p2$ são respectivamente p'_{ref1} e p'_{ref2} e pI^* os ponto de intersecção com as várias linhas.

As condições que definem que a linha não atravessa a área da imagem são as seguintes:

- Se ambos os pontos pertencerem à região esquerda (*bit 1* = 1);
- Se ambos os pontos pertencerem à região direita (*bit 2* = 1);
- Se ambos os pontos pertencerem à região baixo (*bit 3* = 1);

- Se ambos os pontos pertencerem à região cima (bit 4 = 1);

Caso uma das condições referidas se verifique, então a linha não intersecta a área da imagem, logo, não é necessário calcular os pontos de intersecção. Caso ambos os pontos estejam na área 0000 não é necessário efectuar *clipping* porque ambos os pontos são interiores. Caso ambas as condições anteriores não se verifiquem é necessário efectuar o *clipping*, ou seja, verificar onde a linha de referência intersecta as linhas delimitadoras da imagem.

Para efectuar o *clipping* calcula-se inicialmente os quatro pontos de intersecção com as linhas delimitadoras, como demonstrado na figura 39.

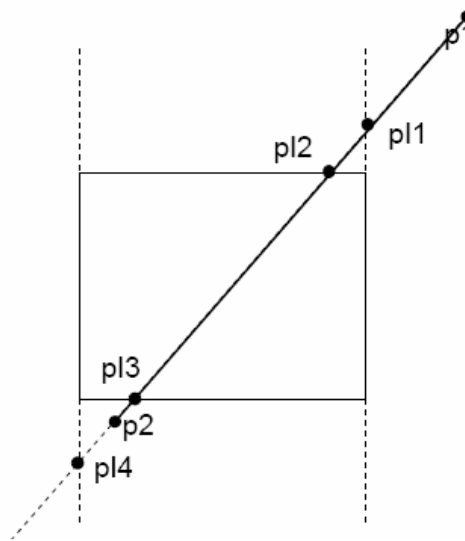


Figura 39: Intersecção da linha de referência com as linhas delimitadoras

Após este cálculo é necessário testar quais destes pontos se encontram efectivamente no interior da área da imagem, no caso apresentado na figura 39, *pl2* e *pl3* são os escolhidos. Tomando em atenção que mais de dois pontos de intersecção podem ocorrer, quando a linha passa por um vértice, neste caso apenas consideramos que ocorre uma única intersecção, ou seja, no máximo teremos sempre dois pontos de intersecção. Tal como referido anteriormente o *clipping* é efectuado se ambos os pontos limites são exteriores ou apenas um único ponto é exterior. No caso de ambos os pontos serem exteriores, então vamos necessariamente ter duas intersecções e esses valores serão utilizados para os novos limites da linha. Caso apenas um ponto limite seja exterior implica que temos de escolher apenas um novo ponto limite. Para efectuar esta escolha é necessário verificar se os pontos de intersecção se encontram no interior de uma área definida pelos pontos limites da recta, como apresentado na figura 40.

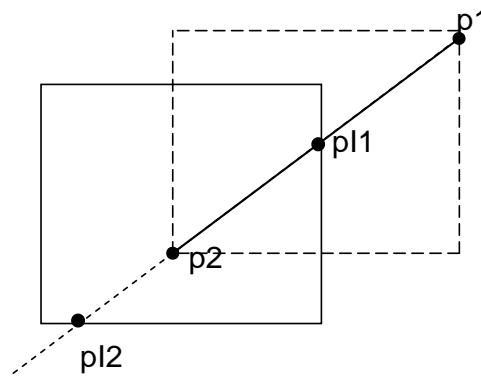


Figura 40: Área para escolha do novo ponto limite da recta de referenciação

Como demonstrado na figura 40 o ponto $p1$ é o escolhido porque encontra-se dentro da área de selecção.

Ou seja o *clipping* consiste em: inicialmente achar os pontos de intersecção interiores e seguidamente, caso apenas um ponto limite da recta seja interior, verificar qual o ponto de intersecção que se encontra dentro da área de selecção.

Após todas estas operações podemos utilizar os dois pontos interiores e desenhar a linha de referência (linha a azul) directamente sobre a imagem de referência, concluindo o processo.

5.5.6 Informação sobre a Imagem

A informação sobre a imagem em análise permite de uma forma imediata, sem recorrer às *Tags* DICOM, visualizar sobre a imagem algumas informações importantes. A informação disponibilizada pode ser observada na figura 41, onde apenas o nome do paciente se encontra escondido. O botão associado é o sexto a contar do topo (circulo vermelho). Para ver a informação basta clicar sobre o botão correspondente.



Figura 41: Exemplo - Informação sobre a imagem

5.5.7 Rotação

A rotação permite que o utilizador efectue rotações à imagem em análise. Os ângulos de rotação são de 90° e a sua direcção é contrária aos ponteiros do relógio. Na Figura 42 podemos ver um exemplo da aplicação de uma rotação. O botão associado é o sétimo a contar do topo. Para efectuar a rotação basta clicar sobre o botão correspondente. Esta alteração é reflectida em todas as imagens da série correspondente.

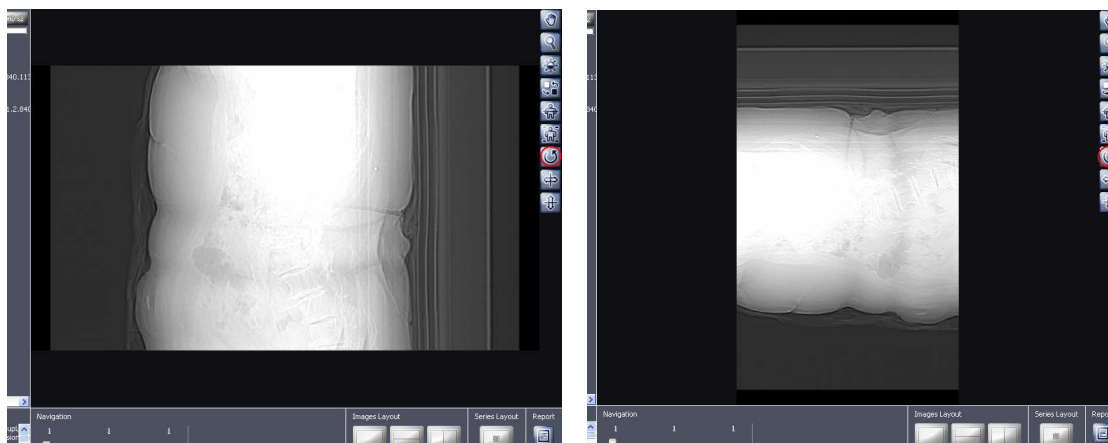


Figura 42: Exemplos – Rotação

5.5.8 Flip Horizontal

O *flip* horizontal permite espelhar a imagem horizontalmente, como apresentado na figura 43. O botão associado é o oitavo a contar do topo. Para efectuar o *flip* basta clicar sobre o botão correspondente. Esta alteração é reflectida em todas as imagens da série correspondente.

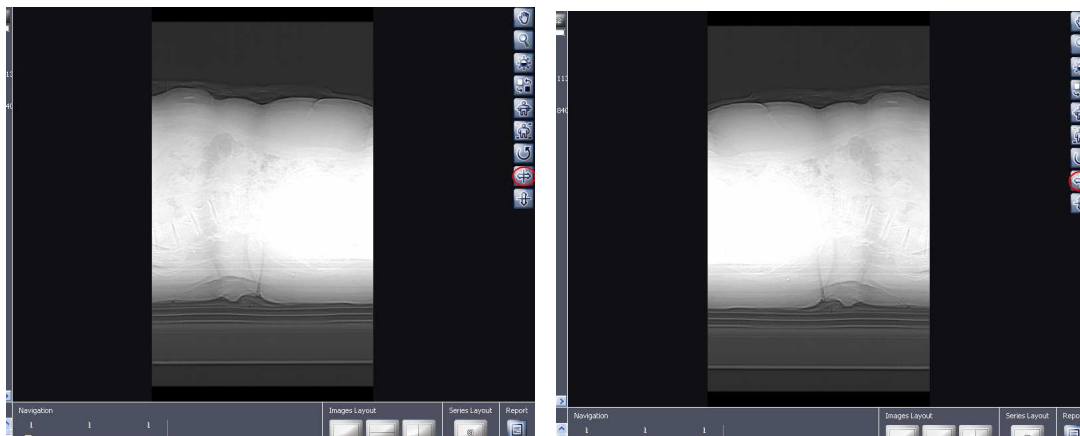


Figura 43: Exemplo – *Flip Horizontal*

5.5.9 Flip Vertical

O *flip* vertical permite espelhar a imagem verticalmente, como apresentado na figura 44. O botão associado é o nono a contar do topo. Para efectuar o *flip* basta clicar sobre o botão correspondente. Esta alteração é reflectida em todas as imagens da série correspondente.

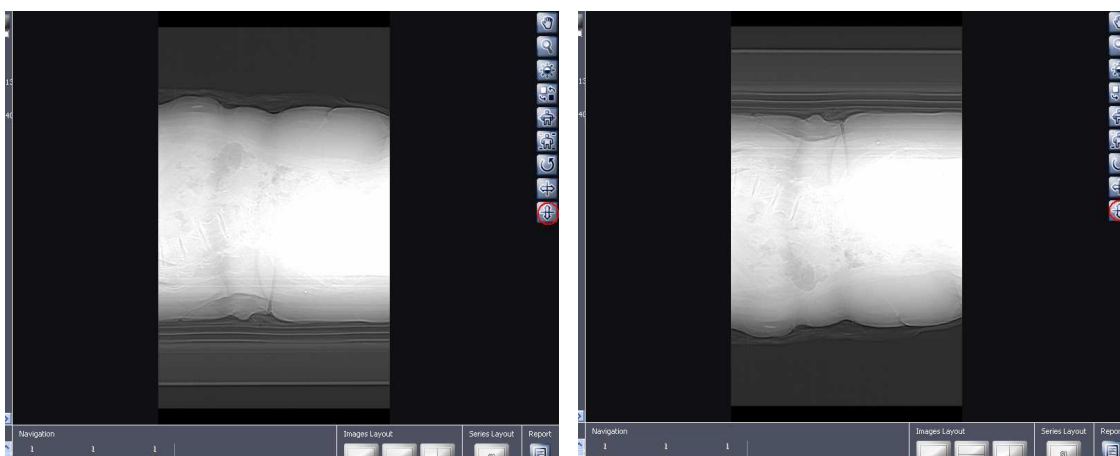


Figura 44: Exemplo – *Flip Vertical*

Capítulo 6

Implementação do Protótipo

A implementação do protótipo, foi sujeita a um conjunto de considerações tecnológicas que serão analisadas neste capítulo. Inicialmente serão abordadas as tecnologias de suporte do sistema, como Linguagens de Programação e APIs (*Application Program Interfaces*), seguindo-se a descrição da utilização do Mapeamento de Texturas utilizando o OpenGL, para visualização das imagens médicas. No final serão discutidas algumas optimizações realizadas para aumentar a eficiência do sistema.

6.1 Tecnologias de suporte

As tecnologias utilizadas foram criteriosamente escolhidas, para garantir que as funcionalidades do protótipo pretendidas pudessem ser implementadas. De seguida são apresentadas as várias tecnologias envolvidas.

6.1.1 Linguagens de programação

As linguagens de programação adoptadas neste trabalho foram:

- Java
- Ansi C/C++

Isto porque as APIs utilizadas dependem destas linguagens de programação.

6.1.2 Interpretação de dados DICOM

A capacidade de interpretação de dados nos formatos de imagem médica, no nosso caso o DICOM, é indispensável porque caso contrário não é possível utilizar a informação correctamente. Felizmente existem APIs que permitem efectuar esta interpretação (*parsing*) e assim gerar informação preparada para que possa ser correctamente utilizada. As APIs, seleccionadas, que permitem efectuar a interpretação de imagens DICOM, são:

- DCMTK – DICOM toolkit [WEB-DCMTK'05]
- PixelMed Java DICOM Toolkit [WEB-PIXELMED'05]

Embora estas duas APIs apresentem funcionalidades semelhantes e são ambas distribuídas gratuitamente, optou-se pela PixelMed porque constatou-se que a documentação é bastante superior.

6.1.3 APIs Gráficas

A API gráfica de baixo nível designada OpenGL [WEB-OPENGL'05] foi utilizada porque esta é considerada a norma de facto, para visualização gráfica. A um nível de abstracção superior encontram-se os GUI (*Graphical User Interface*) *toolkits* que possuem um conjunto predefinido de componentes/objectos gráficos, também designados por *widgets*, que podem ser manipulados. Alguns destes GUIs fornecem objectos onde é possível integrar contextos OpenGL. Numa primeira fase foram pesquisados *toolkits* que possibilitam a utilização do OpenGL. Da pesquisa efectuada foram seleccionados os seguintes GUIs *toolkits*:

- FLTK (Fast Light Toolkit) [WEB-FLTK'05]]
- FOX [WEB-FOX'05]
- Trolltech – Qt [WEB-QT'05]
- wxWidgets [WEB-WXWIDGETS'05]
- GTK+, The GIMP Toolkit [WEB-GTK+'05]

Desta selecção foi excluído à partida o FLTK porque não permite ocultar a barra de comandos do sistema operativo Windows. O segundo a ser eliminado foi o GTK+ porque apesar de possuir um conjunto de funcionalidades muito interessante, revelou-se uma API extremamente instável em particular quando utilizada no sistema operativo Windows. O próximo a ser excluído foi o FOX, porque comparado com o Trolltech – Qt e o wxWidgets a sua documentação é inferior assim como as suas funcionalidades. A escolha final recaiu sobre o Trolltech – Qt e o wxWidgets sendo seleccionado o wxWidgets. O motivo principal pela selecção do wxWidgets deveu-se ao facto do Trolltech – Qt para o sistema operativo Windows apenas possuir uma licença comercial, quando da realização desta pesquisa.

6.1.4 API de comunicação

Devido ao facto do sistema desenvolvido neste trabalho possuir uma componente onde é necessário recorrer a comunicações entre máquinas diferentes, ou seja um sistema distribuído, foi necessário utilizar uma API de comunicação. Contudo constatou-se que o wxWidgets já possui uma API de comunicação via *sockets*. Este facto é sem dúvida uma mais valia, isto porque devido ao wxWidgets ser um *toolkit* multiplataforma não é necessário considerar as especificidades das APIs de comunicação de cada plataforma, porque utilizando o wxWidgets o processo é transparente.

6.2 Mapeamento de texturas utilizando OpenGL

O mapeamento de texturas permite que as imagens sejam correctamente geradas e visualizadas nos contextos OpenGL das interfaces do sistema. Cada imagem médica é efectivamente uma textura que é mapeada a um rectângulo com a dimensão desejada, sendo as áreas onde se encontram as imagens definidas pelo protocolo de afixação, como descrito na secção 5.3 e observado na figura 45. Utilizando as funcionalidades do OpenGL, é possível enviar directamente as texturas para a memória da placa gráfica libertando assim a memória RAM desta informação.



Figura 45: Sistema de visualização *multidisplay* em funcionamento

6.2.1 Criação e visualização das texturas

Quando é efectuado o carregamento de uma série é necessário criar a textura de cada imagem médica antes que estas possam ser visualizadas. Para criar as texturas é necessário utilizar os dados em bruto DICOM (DICOM *raw data*). Nos casos de modalidades como TC e RM, estes dados possuem os valores das densidades dos materiais. Utilizando os valores das densidades podemos calcular a cor de cada ponto das texturas (*texel*). Para tal utiliza-se o pseudocódigo encontrado na norma DICOM, parte 3 - secção 11.2.1.2.

```

se(  $x \leq c - 0.5 - (w-1)/2$  )
{
     $y = y_{min}$ 
}
c.c. se(  $x > c - 0.5 + (w-1)/2$  )
{
     $y = y_{max}$ 
}
c.c
{
     $y = ((x - (c - 0.5)) / (w-1) + 0.5) * (y_{max} - y_{min}) + y_{min}$ 
}

```

onde x representa o valor de entrada (*raw data*), y o valor de saída (índice para a tabela de cor - *Lookup Table*) com uma gama de valores entre y_{min} e y_{max} , c o centro da janela e w a largura da janela. Os valores de saída são índices de uma *lookup table* que contem os valores das cores utilizadas. Os valores limite dos índices para a *lookup table* são: $(y_{min}, y_{max}) = (0, lookup\ table\ size - 1)$. Utilizando estes índices as cores dos *texels* podem ser obtidas directamente.

Contudo não basta só conhecer estes valores é necessário efectuar um conjunto de procedimentos, recorrendo aos comandos OpenGL para podermos visualizar correctamente as texturas, i.e. efectuar o *rendering*. O primeiro passo consiste em gerar a textura e armazenar o valor de *bind* a utilizar, como é apresentado de seguida:

```

glGenTextures(1, &val);
this->bindNumber=val;

```

Seguidamente é necessário efectuar o *bind* da textura, envia-la para a placa gráfica e especificar os filtros utilizados, da seguinte forma:

```

glBindTexture(GL_TEXTURE_2D, this->bindNumber);
glPixelStorei(GL_UNPACK_ALIGNMENT, 1);
getTextDim();
setTextImage(imageRGBA_);

glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_REPEAT);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_REPEAT);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);

```

É necessário dar especial atenção às texturas cujas dimensões (horizontal e vertical) não são potências de 2, porque o OpenGL não consegue utilizar estas texturas. Esta é a razão da existência da função `getTextDim()`. Esta é utilizada para obter a dimensão possível que mais se aproxima da dimensão original da imagem. Após esta operação é efectuado o *set* da imagem utilizando a função `setTextImage(imageRGBA_)` que tem em consideração a dimensão previamente calculada, esta função é também responsável pelo envio da textura para a placa gráfica.

Finalmente o mapeamento e apresentação da imagem são realizados recorrendo ao valor de *bind* da textura, como apresentado de seguida:

```
glBindTexture(GL_TEXTURE_2D, this->bindNumber);

glBegin(GL_QUADS);
    glTexCoord2f(...); glVertex3f(...);
    glTexCoord2f(...); glVertex3f(...);
    glTexCoord2f(...); glVertex3f(...);
    glTexCoord2f(...); glVertex3f(...);
glEnd();
```

6.2.2 Substituição das texturas

Quando o utilizador altera atributos da imagem, como o centro e largura da janela implica que a textura seja actualizada. Tal como referido em [OPENGLREDBOOK'03] no capítulo sobre mapeamento de textura, criar uma textura pode ser computacionalmente mais dispendioso do que modificar uma já existente. Em muitos casos é preferível substituir toda ou parte da textura com nova informação, em vez de criar uma nova.

Foi exactamente este processo de substituição que foi implementado. Inicialmente é necessário verificar se a textura já possui um valor de *bind* atribuído, caso não possua é necessário criar uma textura nova e neste caso não é efectuada a substituição. Caso contrário, i.e. se a textura possui um valor de *bind*, esta é acedida utilizando o valor de *bind* e a substituição é realizada da seguinte forma:

```
glBindTexture(GL_TEXTURE_2D, this->bindNumber);
glTexSubImage2D(...);
```

6.2.3 Eliminação das texturas

Quando é efectuada o descarregamento de uma série todas as texturas correspondentes necessitam de ser eliminadas da memória da placa gráfica. Para efectuar esta eliminação é necessário verificar se o valor de *bind* da textura corresponde efectivamente a uma textura presente na placa gráfica, caso este teste seja verdadeiro a textura é eliminada, como apresentado de seguida:

```
se(glIsTexture((GLuint) this->bindNumber))
{
    glDeleteTextures(1, (const GLuint *) &this->bindNumber);
}
```

6.3 Optimizações

Foram introduzidas algumas optimizações para aumentar a taxa de geração e substituição das imagens (texturas). Na secção anterior referimos que utilizamos substituição de texturas. Ou seja, as texturas são substituídas em vez de: serem geradas, efectuado o carregamento para a placa gráfica e um novo *bind*. Esta constitui a primeira optimização introduzida.

Quando o utilizador altera os parâmetros de uma imagem toda a série é afectada, isto porque, todas as imagens de uma série são dependentes. Devido a este facto foi desenvolvida uma abordagem onde os índices para a obtenção da cor dos *texels*, recorrendo à *lookup table*, são calculados não por imagem mas sim por série. Para tal é necessário verificar inicialmente todos os valores diferentes de densidade existentes na série. Conhecendo estes valores é possível calcular os valores de saída y (secção 6.2.1), i.e. os índices para a *lookup table*. Os objectos imagem, ou seja as instâncias da classe imagem, apenas necessitam de manter índices para a lista de densidades, e não os valores das densidades, porque estes podem ser obtidos através dos índices. Por exemplo se a primeira densidade da imagem (correspondente ao primeiro *texel*) conter a densidade 2000 então é necessário verificar qual a posição/índice deste valor na lista de densidades, que pode por exemplo ser o índice 100 e armazená-lo. Desta forma é possível manter uma lista de densidades por série, em que cada imagem pode utilizar os índices para esta lista para obter os valores respectivos.

Contudo efectivamente não se pretende obter os valores das densidades, mas sim os valores das cores de cada *texel* das imagens. Tal como foi referido anteriormente cada densidade gera um índice para a *lookup table* que é utilizado para obter a cor pretendida. Estes índices encontram-se organizados da mesma forma que a lista de densidades, logo os índices para a lista de densidades presentes nos objectos imagem, podem ser utilizados também para aceder aos valores das cores e desta forma podem ser obtidas as cores associadas a cada *texel*.

Recorrendo a este método é eliminada a computação redundante dos valores dos índices para a *lookup table* (y) que utilizam os mesmos valores de densidade (x), ou seja os valores y são calculados uma única vez. Para reforçar esta ideia, vamos considerar que uma série contém 100 imagens em que todas possuem os mesmos valores de densidade. Não utilizando o método desenvolvido seria necessário computar os valores de y para cada *texel* em todas as imagens, utilizando o nosso método esse cálculo é feito apenas uma vez, porque é utilizada a lista de densidades, correspondendo ao cálculo de apenas uma imagem. Mais, se as densidades dentro de cada imagem possuírem valores repetidos, o que é um caso bastante vulgar, seria necessário calcular repetitivamente os valores de y . Por todos estes factores apontados tornam-se claros os benefícios da utilização do método desenvolvido.

A última optimização introduzida consiste em efectuar a substituição das imagens apenas quando as imagens irão ser apresentadas. Existem três razões para tal, a primeira prende-se com o facto de que para substituir todas as imagens da série ao mesmo tempo seria computacionalmente exigente e o utilizador iria notar, pois a interface poderia congelar enquanto é efectuada a substituição. A segunda razão deve-se ao facto da substituição das imagens ter de ser realizada em tempo real. Por exemplo quando o utilizador move o rato para modificar o centro e largura da janela. Se estas alterações fossem realizadas imediatamente em todas as imagens da série, não seria possível alterar estes parâmetros em tempo real, devido aos custos computacionais envolvidos. E finalmente a última razão deve-se ao facto de que em certos casos o utilizador nunca irá visualizar algumas das imagens da série. Logo não é necessário substituí-las.

Capítulo 7

Conclusões finais e trabalho futuro

Foi apresentado o protótipo do sistema de visualização de imagens médicas *multidisplay*, que permite apoiar os especialistas de imagem médica na navegação e análise dos dados médicos. O protótipo implementa o sistema multiagente, oferecendo a sua flexibilidade e escalabilidade. Permitindo que facilmente se possa montar um sistema que cresça de acordo com as necessidades e recursos dos utilizadores. Pode ainda ser construído utilizando *hardware* convencional, i.e., não é necessário a utilização de placas gráficas especiais (por exemplo com múltiplas saídas) e/ou outro *hardware* específico.

7.1 Conclusões finais

O sistema de visualização é da maior importância para os especialistas de imagem médica. Os mecanismos de navegação e uma área de trabalho de maiores dimensões, proporcionados pelo sistema, facilitam a selecção, visualização e comparação das imagens, auxiliando assim, a analisar rapidamente os dados.

Foi também demonstrado que a estação de controlo satisfaz simultaneamente duas necessidades dos utilizadores. Por um lado permite a navegação e controlo do protocolo de afixação, por outro permite a análise específica de imagens isoladas, utilizando as ferramentas de processamento e análise. Isto graças à forma de navegação por níveis desenvolvida, que sem dúvida é uma das mais valias deste trabalho. Esta é uma característica única, não encontrada nos outros visualizadores, permitindo de uma forma rápida e intuitiva mudar a percepção que se possui da informação, podendo-se estabelecer um paralelismo com a alteração de níveis de detalhe encontrada na área da computação gráfica. Ou seja quanto mais elevado o nível, maior o detalhe, sendo o nível superior o da imagem.

A interface da estação de controlo foi desenvolvida com o auxílio de um designer gráfico, para garantir, uma interface agradável e intuitiva. Na estação de controlo toda a informação encontra-se presente directamente na sua interface, não existindo a utilização de sub-janelas. Desta forma o utilizador rapidamente se familiariza com o sistema e encontra as funcionalidades que deseja. Estas funcionalidades encontram-se delimitadas por áreas que são facilmente identificadas por informação textual. Foram ainda desenvolvidos botões com *icons* que reflectem as funcionalidades associadas. Desta forma o resultado é uma ferramenta extremamente intuitiva e de fácil aprendizagem.

As optimizações introduzidas (secção 6.3), permitiram que o sistema funcione de uma forma harmonizada, i.e., sem que o utilizador perceba a carga computacional envolvida na actualização das imagens. Através destas optimizações, garante-se também a eficiência do sistema, porque o tempo de análise pode ser reduzido devido à actualização mais rápida das imagens.

O protótipo existente foi desenvolvido para o sistema operativo Windows. Contudo devido às tecnologias utilizadas facilmente se podem obter versões para outros sistemas operativos. É de salientar que o protótipo pode também, operar num sistema heterogéneo, constituído por exemplo: por máquinas com os sistemas operativos Windows e Linux.

Devido à versatilidade do sistema, garantida pelo sistema multiagente, poderiam ser considerados outros cenários de aplicação para além da área da medicina. Para tal seria necessário efectuar pequenas alterações (por exemplo considerar outros formatos de imagem), no entanto os mesmos conceitos podem ser aplicados. Por exemplo, este sistema poderia ser utilizado no contexto publicitário. Considerando que o cenário é uma loja que possui vários terminais de visualização (*displays*) e ainda uma estação de controlo operada pelo responsável pelos conteúdos publicitários disponibilizados, este poderia alterar as imagens/conteúdos de todos os *displays* através da estação de controlo. O que sem dúvida oferece uma mais valia considerável, pois a tarefa de alteração de conteúdos seria extremamente facilitada. Contudo muitos outros cenários e áreas de aplicação podem ser considerados, devido às características do sistema.

Constatou-se, através do protótipo desenvolvido, que os agentes ou sistemas multiagente são uma tecnologia que traz uma grande quantidade de mais valias para este tipo de sistemas. Nomeadamente a escalabilidade, que no nosso caso, permite a utilização de um número variado de *displays*. E o facto de podermos adicionar agentes, que desempenham tarefas diversas, fornece uma grande versatilidade ao sistema. Este é um factor que poderia ser explorado de uma forma mais aprofundada pelos sistemas de RV actuais, isto se for considerado de raiz um sistema multiagente para visualização, controlo, etc. do sistema.

7.2 Trabalho futuro

Presentemente todo o sistema encontra-se a ser avaliado, e temos esperança que no futuro próximas versões de teste possam ser introduzidas nas instalações de instituições de imagem médica, permitindo assim um *feedback* valioso por parte dos especialistas.

É necessário continuar o desenvolvimento das ferramentas de processamento e análise de imagem. Isto porque as ferramentas desenvolvidas representam um conjunto básico, sendo necessário introduzir ferramentas, como por exemplo as medições.

Será interessante considerar e propor, formas de navegação alternativas. Apesar do método de navegação desenvolvido ser considerado uma mais valia significativa, seria interessante compará-lo com outros métodos alternativos. Para tal é necessária a colaboração dos especialistas, para fornecerem dados que permitam avaliar as diversas soluções.

O sistema multiagente poderá sofrer alterações, com a introdução de novos agentes que implementem ferramentas de apoio à decisão, tais como diagnóstico assistido por computador e protocolo de afixação inteligente. Em especial para o segundo exemplo, poderão ser desenvolvidos agentes de conhecimento que monitorizam as acções do utilizador formando assim uma base de conhecimento, que poderá ser utilizada para inferir qual o protocolo de afixação mais apropriado. Por exemplo se um determinado especialista utiliza constantemente o mesmo protocolo de afixação para uma determinada modalidade, então faz algum sentido que sempre que um novo exame for analisado, o protocolo de afixação do sistema seja o que ele costuma utilizar. Como se pode constatar, existem muitas varáveis a serem consideradas, ou seja o protocolo de afixação depende do especialista, da modalidade, etc. A introdução destes novos agentes é facilitada pela arquitectura multiagente desenvolvida.

Bibliografia

- [ALVES'02] Alves, V. M. R., **Resolução de Problemas em Ambientes Distribuídos. Uma Contribuição nas Áreas da Inteligência Artificial e da Saúde**, Tese de Doutorado, Universidade do Minho, 2002.
- [BIMBER et al.'01] Bimber, O., Fröhlich, B., Schmalstieg, D., and Encarnação, L.M.. **The Virtual Showcase**, IEEE Computer Graphics & Applications, vol. 21, no. 6, pp. 48-55, 2001.
- [BIMBER et al.'02] Bimber, O., Gatesy, S.M., Witmer, L.M., Raskar, R. and Encarnação, L.M.. **Merging Fossil Specimens with Computer-Generated Information**, in IEEE Computer, September, pp. 45-50, 2002.
- [BLOCH et al.'46] Bloch F, Hanson WW, Packard M. **Nuclear induction**. Phys Rev 1946; 69: 127.
- [BRUCE'03] Bruce, Robert, **RIS/PACS integration – what is it and what are its benefits?**, AuntMinnie, 2003.
- [CAVEDON&TILHAR'95] Cavedon, L., Tilhar, G., **A Logical Framework for Multiagents Systems and Joint Attitudes**, in Proceedings of First Australian Workshop on Distributed Artificial Intelligence, Canberra, Australia, 1995.
- [DEMAZEAU&MÜLLER'90] Demazeau, Yves; Müller, Jean Pierre. **Decentralized Artificial Intelligence**. In: Decentralized Artificial Intelligence - 1. Org.: Demazeau, Yves, Müller, Jean-Pierre. North-Holland, Elsevier Science Publishers, 1990. p. 3ss.
- [FAN et al.'02] Fan, Yong, Jiang, Tianzi, Evans, David J., **Volumetric Segmentation of Brain Images Using Parallel Genetic Algorithms**, IEEE Transactions on Medical Imaging, Vol 21, No. 8, pp. 904-909, August 2002
- [FRANKLIN&GRASSER'96] Franklin, Stan, Graesser, Art, **Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents**, Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages, Springer-Verlag, 1996
- [FERBER&GASSER'91] Ferber, J.; Gasser, L. **Intelligence artificielle distribuée**. In: International Workshop on expert systems & their applications, 10., 1991, Avignon. Cours n. 9. France: [s.n], 1991.

- [KINDLMAN&DURKIN'98] Kindlman, G. and Durkin, J. **Semi-Automatic Generation of Transfer Functions for Direct Volume Rendering**, *Proc.1998 IEEE Symp. Volume Visualization*, IEEE CS Press, Los Alamitos, Calif., Oct. 1998, pp. 79-86.
- [KUEDERLE et al.'01a] Kuederle, O., Inkpen, K., Atkins, M.S., **Evaluation of viewing methods for magnetic resonance images**, *Proceedings of the SPIE-Medical Imaging 2001*, (4319): 536-545, Feb. 2001.
- [KUEDERLE et al.'01b] Kuederle, O., Inkpen, K., Carpendale, M.S.T., Atkins, M.S., **Interacting with image sequences: Detail-in-context and thumbnails**, *Proceedings of Graphics Interface 2001*, pp 111-118, Jun. 2001.
- [MCINERNEY&TERZOPOULOS'99] McInerney, Tim and Terzopoulos, Demetri; **Topology Adaptive Deformable Surfaces for Medical Image Volume Segmentation**; *IEEE Transactions on Medical Imaging*, Vol 18, No. 10, pp. 840-850, October 1999.
- [MANSFIELD&MAUDSLEY'76a] Mansfield P, Maudsley AA. **Planar spin imaging by NMR**. *J Phys C: Solid State Phys* 1976; 9: L409-411.
- [MANSFIELD&MAUDSLEY'76b] Mansfield P, Maudsley AA. **Line scan proton spin imaging in biological structures by NMR**. *Phys Med Biol* 1976; 21: 847-852.
- [MARREIROS'02] Marreiros, Filipe M., **Realtime Direct Volume Rendering**, Relatório Técnico, Universidade do Algarve, 2002.
- [MARREIROS et al.'05a] Marreiros, F., Figueiredo, P., Pereira, P., Matos, N., Marcos, A.. **An augmented reality showcase to support a cultural heritage scenario**, In: *Proc. of Training, Education & Simulation International (TESI) 2005*, Maastricht, 22-24 March 2005, Netherlands.
- [MARREIROS et al.'05b] Marreiros, F., Marcos, A.. **Calculating the Stereo Pairs of a Mirror-Based Augmented Reality System**, In: *Proc. of 13º EPGC*, Vila Real, 12-15 October, 2005, Portugal.
- [MOULIN&CHAIB-DRAA'96] Moulin, Bernhard and Chaib-Draa, Brahim. **An overview of distributed artificial intelligence**, In Greg M. P. O'Hare and Nicholas R. Jennings, editors, *Fundamentals of Distributed Artificial Intelligence*, pages 3-56, John Wiley and Sons, 1996.
- [NEVES&MACHADO'97] Neves, J., Machado, J., **Formalizing Context in Knowledge Management Systems**, in *Proceedings of the International Workshop of Distributed*

Artificial Intelligence and Multi-Agent Systems (DAIMAS'97), St Petersburg, Russia, 1997.

[OLIVEIRA'96] Oliveira, F.M.. **Inteligência Artificial distribuída**. In IV Escola Regional de Informática, Londrina-Itajaí-Canoas, Brasil, Abril.

[OPENGLREDBOOK'03] Shreiner, D., Woo, M., Neider, J., Davis, T., The **OpenGL Programming Guide - The Redbook: The Official Guide to Learning OpenGL**, Version 1.4, Fourth edition, OpenGL Review Board, Addison-Wesley, 2003.

[PFISTER et al.'01] Pfister, H., Lorensen, B., Bajaj, C., Kindlmann, G., Schroeder, W., Avila, L., Raghu, K., Machiraju, R., and Lee, J., **The transfer function bake-off**, IEEE Computer Graphics and Applications 21, pp. 16-22, Jan-Feb 2001.

[PURCELL et al.'46] Purcell EM, Torrey HC, Pound RV. **Resonance absorption by nuclear magnetic moments in a solid**. Phys Rev 1946; 69: 37-38.

[ROCHA et al.'03] Rocha, A. R., Júnior, E. M. S. , Alves, J. C.. **Introdução aos Agentes Inteligentes e aos Sistemas Multiagentes**, Universidade Federal de Lavras, Departamento de Ciências da Computação, 6 Abril, 2003.

[SANTOS'99] Santos, M., **Sistemas de Classificação em Ambientes Distribuídos**, Tese de Doutorado, Universidade do Minho, Braga, Portugal, 1999.

[SICHMAN et al.'92] Sichman Jaime Simão; Demazeu, Yves; Boissier, Olivier. **How can knowledge-based systems be called agents?** In: Simpósio Brasileiro de Inteligência Artificial, 9, 1992, Rio de Janeiro. Anais... Rio de Janeiro: Pontifícia Universidade Católica do Rio de Janeiro, 1992. p.173-185.

[SCHIKORE et al.'00] Schikore, D.R., Fischer, R.A., Frank, R., Gaunt, R., Hobson, J., Whitlock, B. **High-Resolution Multiprojector Display Walls**. IEEE Computer Graphics and Applications, Vol. 20(4) (2000) 38-44

[TORY et al.'05] Tory, Melanie; Simeon, Potts and Möller, Torsten; **A Parallel Coordinates Style Interface for Exploratory Volume Visualization**; IEEE Transactions on Visualization and Computer Graphics, Vol 11, No. 1, pp. 71-80, Jan-Feb 2005.

[TSCHIRLEY et al.'03] Tschirley, René, Köchy, Kai, Mäle, Steffen, **Integrating patient-oriented data processing into the PREPaRe virtual hospital using XML technology**, CARS 2003. Computer Assisted Radiology and Surgery. Proceedings of

the 17th International Congress and Exhibition, London, June 25-28, 2003, pp 259-264.

[VAN DER HEYDEN'98a] van der Heyden, J. E., **Magnetic resonance image viewing and the "Screen real estate problem"**, Msc. Thesis, Simon Fraser University, 1998.

[VAN DER HEYDEN et al.'98b] van der Heyden, J. E., Carpendale, M.S.T., Inkpen, K., Atkins, M.S., **Visual Representation of Magnetic Resonance Images**, Proceedings Vis98, 1998, pp 423-426.

[VAN DER HEYDEN et al.'99a] van der Heyden, J. E., Atkins, M.S., Inkpen K., Carpendale, M.S.T., **MR image viewing and the screen real estate problem**, *Proceedings of the SPIE-Medical Imaging 1999*, 3658: 370-381, Feb. 1999.

[VAN DER HEYDEN et al.'99b] van der Heyden, J. E., Atkins, M.S., Inkpen K., Carpendale, M.S.T., **A User Centered Task Analysis of Interface Requirements for MRI Viewing**, Graphics Interface 1999, pp 18-26.

Referências na Web

[WEB-ACR'05] <http://www.acr.org>. Acedido em 27-03-2005.

[WEB-ANSI'05] <http://www.ansi.org>. Acedido em 13-04-2005.

[WEB-CHROMIUM'06] <http://chromium.sourceforge.net>. Acedido em 10-01-2006.

[WEB-DCMTK'05] <http://dicom.offis.de/dcmtdk.php.en>. Acedido em 07-05-2005.

[WEB-DICOM'05] <http://www.psychology.nottingham.ac.uk/staff/cr1/dicom.html>. Acedido em 23-02-2005.

[WEB-DICOMWORKS'05] <http://dicom.online.fr>. Acedido em 12-04-2005.

[WEB-EFILM'05] <http://www.merge.com/index.asp>. Acedido em 08-04-2005.

[WEB-FLTK'05] <http://www.fltk.org>. Acedido em 18-05-2005.

[WEB-FOX'05] <http://www.fox-toolkit.org>. Acedido em 03-06-2005.

[WEB-GTK+'05] <http://www.gtk.org>. Acedido em 13-05-2005.

[WEB-HL7'05] <http://www.hl7.org>. Acedido em 11-03-2005.

[WEB-IMAGEJ'05] <http://rsb.info.nih.gov/ij>. Acedido em 02-03-2005.

[WEB-JULIUS'05] <http://www.julius.caesar.de>. Acedido em 22-06-2005.

[WEB-LINE-LINEINTERSECTION'05] <http://mathworld.wolfram.com/Line-LineIntersection.html>. Acedido em 18-11-2005.

[WEB-MRICO'05] <http://www.psychology.nottingham.ac.uk/staff/cr1/mrico.html>. Acedido em 09-03-2005.

[WEB-NEMA'05] <http://www.nema.org>. Acedido em 10-07-2005.

[WEB-OPENGL'05] <http://www.opengl.org>. Acedido em 24-02-2005.

[WEB-OSIRIS'05] http://www.sim.hcuge.ch/osiris/01_Osiris_Presentation_EN.htm. Acedido em 17-05-2005.

- [WEB-PLANE-PLANEINTERSECTION'05] <http://mathworld.wolfram.com/Plane-PlaneIntersection.html>. Acedido em 16-11-2005.
- [WEB-PIXELMED'05] <http://www.pixelmed.com>. Acedido em 11-04-2005.
- [WEB-POINT-LINEDISTANCE2-DIMENSIONAL'05] <http://mathworld.wolfram.com/Point-LineDistance2-Dimensional.html>. Acedido em 18-11-2005.
- [WEB-QT'05] <http://www.trolltech.com>. Acedido em 20-05-2005.
- [WEB-REALTIMESOFT'05] http://www.realtimesoft.com/multimon/reviews/sys_quad. Acedido em 13-08-2005.
- [WEB-RSNA'05] <http://www.rsna.com>. Acedido em 24-07-2005.
- [WEB-RUBOMEDICAL'05] <http://www.rubomedical.com>. Acedido em 15-03-2005.
- [WEB-STEREO3D'05] <http://www.stereo3d.com/3dhome.htm>. Acedido em 29-08-2005.
- [WEB-VOLDG'05] <http://www.merl.com/projects/dg4vg>. Acedido em 21-05-2005.
- [WEB-VRJUGGLER'06] <http://www.vrjuggler.org>. Acedido em 04-01-2006.
- [WEB-WIREGL'06] <http://graphics.stanford.edu/software/wiregl>. Acedido em 06-01-2005.
- [WEB-WXWIDGETS'05] <http://www.wxwidgets.org>. Acedido em 25-02-2005.